University of Waterloo

**David R. Cheriton
School of
Computer Science**

University of Waterloo
200 University Ave. W.
Waterloo, Ontario N2L 3G1
Canada

Telephone: 1-519-885-1211 x37508
Facsimile: 1-519-885-1208
E-mail: trefler@cs.uwaterloo.ca

September 26, 2019

Dear Professor Trlifaj, Professor Hajic, members of the Habilitation Board,

I am writing to strongly recommend that Pavel Parizek, Ph.D, be awarded his Habilitation and appointed as an associate professor.

I am an Associate Professor of the David R. Cheriton School of Computer Science, at the University of Waterloo. My Ph.d. thesis was on efficient model checking and analysis procedures for temporal logics and quantitative temporal logics. Over the last 20 years my research has focussed on the design of program analysis techniques that eliminate or mitigate the 'state explosion problem' in model checking. State explosion refers to the enormous state spaces generated from textual program description. Specifically, I design compositional reasoning techniques that show how to decompose an analysis problem involving several program components into related problems over the individual components. In addition, I have described local symmetry techniques that allow, through systematic compositional reasoning, the analysis of programs composed of many similar components to be carried out on a small numbers of 'representative' components, thereby avoiding the difficulties of state explosion.

Dr. Parizek's work has followed two general directions: first, the use of partial order reduction as a means of fighting state explosion in program analysis; and second, the development of tools to efficiently find program bugs and concurrency errors in multiprocess programs and protocols. Both areas of research are current areas of substantial interest in the model checking and program analysis community and Dr. Parizek's results all appear in highly respected, highly competitive and widely known international research venues, including TACAS, ASE, VMCAI, FMCAD, SPIN and well known journals including SCC and STTT.

Below, I briefly describe a few of Dr. Parizek's specific contributions.

Model checkers offer proof that programs satisfy properties given as propositional temporal specifications. However, in typical model checking applications, it is the instances when a model checker fails to prove that a program satisfies a given specification that the results are most compelling. That is, such a failure is in fact a proof that the program has a bug, or error, and as such the program bug must be fixed. Dr. Parizek's work, *Efficient Detection of Errors in Java Components Using Random Environment and Restarts* published at TACAS in 2010, is of particular interest. The work uses an abstract environmental process to mimic the neighboring processes a particular component may interact with. Being able to detect errors that occur 'local' to the component without having to build the order of magnitude larger environment the component may operate in is particularly useful. This work encodes an ingenious technique for allowing the analysis of complex, distributed programs with a reasonable cost.

Many different techniques have been proposed to mitigate state explosion in model checking, and while it is unlikely that any single technique will suffice for all applications so called 'partial order reduction' techniques have played an important role. Dr. Parizek's paper *Hybrid Analysis for Partial Order Reduction of Programs with Arrays* offers a particularly interesting extension to the basic technique. Partial order techniques reduce the number of computations that a verification tool analyzes by considering as equivalent two computations that differ only in the non-deterministic choice in the order that two threads access a shared object. Using a careful analysis on the access of array elements Dr. Parizek work shows how it is possible to use partial order reduction techniques even when program threads are accessing dynamic objects such as the heap. This is an important extension of the partial order reduction technique as it allows significantly more reduction to multi-threaded dynamic program elements, which are exactly the kind of program elements that have traditionally been difficult to implement without errors. Therefore this technique, and its related analysis, have the potential to have long term impact in the area.

In his paper, *Hybrid Partial Order Reduction with Under-Approximate Dynamic Points-to and Determinacy Information*, Dr. Parizek combines a dynamic partial order analysis with a hybrid field access analysis that results in a new approach to partial order analysis. The new combined approach is applicable in cases not covered by earlier approaches and therefore significantly enhances the ability of partial order techniques to reduced the cost of program analysis. The key point of the new technique is a novel combination of under approximate dynamic points to information in the on-the-fly analysis of program states. Care is then taken, using the hybrid partial order reduction, to ensure that state space traversal covers all interleavings of possibly interfering actions. This involves a deep analysis of the interplay between program components and represents a significant achievement in research into automated program analysis.

In summary, I am very pleased to strongly recommend that Pavel Parizek be awarded his Habilitation and appointed as an associate professor. If you have any questions please do not hesitate to contact me at (519) 885-1211, ext. 37508, or by email at trefler@cs.uwaterloo.ca.

Sincerely,

[redacted]

Richard Trefler
Associate Professor