Formal Techniques for Java-like Programs (FTfJP)

Elvira Albert[1], Anindya Banerjee[2], Sophia Drossopoulou[3], Marieke Huisman[4⋆], Atsushi Igarashi[5], Gary T. Leavens[6], Peter Müller[7], and Tobias Wrigstad[8]

[1] Complutense University of Madrid, Spain
[2] Kansas State University, USA
[3] Imperial College London, UK
[4] University of Twente, Netherlands
[5] Kyoto University, Japan
[6] University of Central Florida, USA
[7] ETH Zurich, Switzerland
[8] Purdue University, USA

Abstract This report gives an overview of the 10th Workshop on Formal Techniques for Java-like Programs at ECOOP 2008. It explains the motivation for the workshop, and summarizes the presentations and discussions.

1 Introduction

Formal techniques can help analyze programs, precisely describe program behavior, and verify program properties. Newer languages such as Java and C# provide good platforms to bridge the gap between formal techniques and practical program development, because of their reasonably clear semantics and standardized libraries. Moreover, these languages are interesting targets for formal techniques, because the novel paradigm for program deployment introduced with Java, with its improved portability and mobility, opens up new possibilities for abuse and causes concern about security.

Work on formal techniques and tools for programs and work on the formal underpinnings of programming languages themselves naturally complement each other. This workshop aims to bring together people working in both these fields, on topics such as: program verification, formal models and extensions of Java-like languages, program analysis, and type systems.

The workshop was organized by Marieke Huisman (INRIA Sophia Antipolis, France), Sophia Drossopoulou (Imperial College London, UK), Susan Eisenbach (Imperial College London, UK), Gary T. Leavens (University of Central Florida, USA), Peter Müller (Microsoft Research, Redmond, USA), Arnd Poetzsch-Heffter (University of Kaiserslautern, Germany), and Erik Poll (Radboud University Nijmegen, Netherlands). The selection of papers was done by a larger

⋆ Affiliated with INRIA Sophia Antipolis at the time of the workshop.

program committee chaired by Marieke Huisman. The committee members are listed at the end of this report.

Around 40 people attended this full-day workshop. A representative list of participants is given at the end of this report. A number of other participants dropped by for specific presentations, to chat with particular speakers, etc. To encourage cross-fertilization with related research areas, the IWACO and FTfJP workshops organized a joint workshop dinner.

Overview of the presented papers. Sixteen research papers were submitted, of which eleven were accepted for presentation at the workshop. The program committee made its selection after a fruitful discussion. Besides quality of the submission, also potential interest of the presentation for the workshop participants was used as a criterion.

The accepted papers are collected in informal proceedings that are available as technical report ICIS-R08013 from the Radboud University Nijmegen, Netherlands, available at `http://www.cs.ru.nl/~erikpoll/ftfjp/FTfJP08`.

The topics addressed by the presented papers are:

- program verification;
- formal models and extensions for Java-like languages;
- program analysis; and
- type systems.

For each topic, the sections below briefly describe the presentations and discussions.

2 Program Verification

Jan Smans talked about joint research with Bart Jacobs and Frank Piessens on the verification of implicit dynamic frames. Dynamic frames are a powerful mechanism for modular verification. They propose a technique that avoids the need to explicitly specify and verify frame conditions; these are replaced by accessibility predicates from which an upper bound on the set of locations that may be modified can be inferred. The technique has been implemented in a tool set, and Jan demonstrated how it could be used to verify several challenging examples. The discussion following the presentation revolved around the similarities with Banerjee et al.'s work on regional logic which was to be presented in the following days as a part of ECOOP's technical track. In regional logic, region expressions can be used to explicitly specify read and write effects that, similar to dynamic frames, needs to be checked at verification time. Implicit dynamic frames do not require these explicit annotations but rely on inferring frame information from preconditions. Finally, the discussion also touched briefly on the subject of patterns, in particular the application to examples involving more layers of structure, e.g. the Composite pattern, which would be useful to test the practical usability of the proposed tool set. However, this did not arrive at any conclusion.

Romain Bardou presented a way to reason about pointer arithmetic and memory separation for low-level languages, based on ownership systems. Because of the low-level language features that are supported by his approach, the verification technique can be applied to C programs. In the following discussion, Dino DiStefano and others questioned whether the assumption of fresh pointer locations was sound in the presence of pointers and pointer arithmetic and that in a C program, a newly allocated object may be given an address already pointed to by preexisting variables. Currently, Bardou's simple formalisation does not model memory deallocation.

The last talk in this session was given by Dave Cunningham, who presented joint work with Susan Eisenbach and Sophia Drossopoulou on the formalization of a lock inference algorithm. A good way to structure concurrent programs (and thus make them less error-prone) is the use of atomic sections. This is a high-level primitive, which can be compiled into transactional memory accesses or a locking schema. This paper discusses an efficient and precise algorithm that infers locks from atomic sections. The algorithm is formalized in Isabelle/HOL and proven correct. The discussion evolved around the possibility to combine lock inference with partial program annotations.

3 Formal models and extensions of Java-like languages

John Boyland presented a new style operational semantics for a concurrent language with fork-join parallelism, synchronization, and volatile fields. The operational semantics introduces the notion of "write-key", which simulates the happens before order of relaxed memory models, i.e., it indicates whether a certain write could happen based on what happened before in the program. The paper then shows that exhibiting a write-key error in the operational semantics is equivalent to the program containing a data race. The advantage of this approach is that write-key errors can be detected locally, whereas data races cannot. The operational semantics and equivalence proof are formalized using Twelf. The discussion mainly focused on issues about the correctness results. There was one question which clarified that correctness holds for any possible execution and not only for a given entry. Also, another question made clear that there is no order required on the write-keys and neither are time-stamps.

Next, Gabriele Costa proposed an extension of Java's security model that would allow to specify, analyze, and enforce history-based security policies. This is joint work with Massimo Bartoletti, Pierpaolo Degano, Fabio Martinelli, and Roberto Zunino. Crucial to the approach is that the policies are local, which makes them easier to enforce and allows for safe composition of programs and their security requirements. This paper designs a run-time mechanism for the enforcement of local history-based security properties, and then further optimizes this, based on a static analysis that detects when a policy might be violated—and thus allows one to discard checks that never fail. During the discussion, Gabriele explained that the models which are obtained for the policies are finite

with respect to the number of states. He also clarified that the expressiveness of their approach is comparable to other history-based approaches.

The last paper in this session was presented by Tetsuo Kamina. He discussed joint work with Tetuso Tamai on a small core language that formalizes key concepts of object adaptability, i.e., the ability of an object to change its behavior dynamically. The small core language is compared with the earlier proposed Epsilon model for object adaptability, and it turns out to be an appropriate formal base for this model .After the presentation, there were several suggestions to improve this work by various workshop participants. One comment was how to handle multiplicity by giving rules for valid multiplicity such that if they are violated then the program is not valid. Another suggestion was to use a typed execution model. Further the relation of this work with roles was discussed, and also, how one could handle the situation where a role defined in a subclass imposes constraints on roles as inherited from superclasses.

4 Program Analysis

The next session started with Elvira Albert presenting joint work with Puri Arenas, Samir Genaim, and Germán Puebla on the handling of numeric fields to automatically prove termination of programs written in a Java-like language. Statistics have revealed that in the Java libraries for over 10% of the loops, termination depends on the values stored in numeric fields. The presentation gave an overview of different program patterns where termination depends on numeric fields, and it sketched how termination proofs for these programs could be found automatically. The discussion evolved on the precision of the analysis, the merits of performing analysis at byte-code or source code level, and a comparison with any optimizations performed by the Java compiler, in particular whether the transformation of field accesses to local variables is already done by the Java compiler.

Next, Rok Strniša presented his work on the Java module system. He analyzed and formalised the core of two JSRs that propose a new module system for Java (which will be part of Java 7). The analysis revealed several shortcomings in the proposal, w.r.t. module instantiations and class resolution. The presentation further proposed clean solutions to these problems, that are also modeled formally (using Isabelle/HOL). This allowed him to prove type soundness for the corrected version of the module system. The discussion evolved on the modifications to the module system suggested by Rok, the practical ramifications of the proposed solutions, and in particular in how far these modifications would be agreeable to the Java community. We also discussed the role of the formal model in discovering these shortcomings.

Last, Samir Genaim presented joint work with Fausto Spoto on the detection of purity of method arguments, by means of an abstract domain where "constancy" is defined as an abstract interpretation. The presentation concluded with examples of how constancy information can be used to improve the precision of other, existing static analyses. The discussion centered around the comparison

of static analyses based on constancy with effect systems based approaches to constancy.

5 Types

The last session started with Alexander Summers presenting joint work with Sophia Drossopoulou and Peter Müller on a Universe-Type based verification technique for static fields and methods. In particular, he discussed how the use of Universe Types for the verification of invariants should be adapted for a language that contains static fields and methods. This required to extend Universe Type hierarchy such that each ownership tree is rooted in a class. This allows classes to own object instances as their static fields. Furthermore, methods need to be annotated by the classes whose static methods they may (directly or indirectly) invoke. These annotations can be reduced by organizing classes in layers. The presentation was followed by a discussion of whether the approach can be made more lightweight by inferring the levels of classes, and the annotations. Also, there was a question whether partial, instead of linear, orders could be used for the partitions of classes; the authors conjectured partial orders could be used.

The last presentation of the workshop was given by Stefan Wehr, who presented joint work with Peter Thieman on subtyping existential types. Existential types are often advocated as a powerful feature that can subsume Java's interface and wildcard types, and several proposals exist to extend Java-like languages with existential types. However, Stefan showed that existential types do not mingle well with subtyping, and make type checking undecidable. He concluded with some possible compromises that allow most of the features of existential types, but keep the subtyping relation decidable. The following discussion centered on the implication of the work on decidability for Java wildcards, and newer applications of existential types into ownership types.

6 Conclusions

A special issue for FTfJP 2008 will appear in the Journal of Object Technology (JOT).

This was the tenth workshop in the series, and the workshop is still going strong. The focus of the workshop has shifted somewhat over time, as different topics become more or less popular, or essentially resolved, while others have gained importance. Moreover, the revival of IWACO (International Workshop on Aliasing, Confinement and Ownership in object-oriented programming) has also contributed to this shift. It is nice to observe that the workshop has helped in raising some interesting topics for research, and to observe the way it has contributed to fostering collaborations, all of which has resulted in good work presented not just at this workshop but also at the main ECOOP conference.

The workshop has somewhat outgrown the standard workshop format, given the number and quality of submissions it typically received, and the number of people that want to participate. But the interest it generates and the audience

it attracts proves that it clearly serves a useful purpose and we look forward to organizing another FTfJP workshop at next year's ECOOP.

Program committee

Elvira Albert, Complutense University of Madrid (Spain)
Cyrille Artho, RCIS/AIST (Japan)
Anindya Banerjee, Kansas State University (USA)
Mike Barnett, Microsoft Research, Redmond (USA)
Amy Felty, University of Ottawa (Canada)
Paola Giannini, University of Eastern Piedmont (Italy)
Rene Rydhof Hansen, Aalborg University (Denmark)
Marieke Huisman (chair), INRIA Sophia Antipolis (France)
Atsushi Igarashi, Kyoto University (Japan)
Bart Jacobs, University of Leuven (Belgium)
Gerwin Klein, National ICT Australia (Australia)
Neelakantan R. Krishnaswami, Carnegie Mellon University (USA)
Matthew Parkinson, University of Cambridge (UK)
Arnd Poetzsch-Heffter, University of Kaiserslautern (Germany)
Tobias Wrigstad, Purdue University (USA)

List of participants

Suad Alagic, University of Southern Maine (USA)
Elvira Albert, Complutense University of Madrid (Spain)
Jonathan Aldrich, Carnegie Mellon University (USA)
Anindya Banerjee, Kansas State University (USA)
Romain Bardou, INRIA Saclay (France)
Massimo Bartoletti, Universitá di Pisa (Italy)
Frederic Besson, IRISA/INRIA (France)
John Boyland, University of Wisconsin-Milwaukee (USA)
Nicholas Cameron, Imperial College (UK)
Dave Clarke, CWI (The Netherlands)
David Cunningham, Imperial College (UK)
Dino Distefano, University of Cambridge (UK)
Sophia Drossopoulou, Imperial College (UK)
Patrick Eugster, Purdue University (USA)
Adrian Fiech, Memorial University (Canada)
Samir Genaim, Technical University of Madrid (Spain)
Paola Gianini, Alessandria (Italy)
Christian Haack, Radboud University Nijmegen (The Netherlands)
Clement Hurlin, INRIA (France)
Atsushi Igarashi, Kyoto University (Japan)
Tetsuo Kamina, The University of Tokyo (Japan)
Gary T. Leavens, University of Central Florida (USA)

Yu David Liu, The Johns Hopkins University (USA)
Nicholas Matsakis, ETH Zurich (Switzerland)
Ana Milanova, Rensselaer Polytechnic Institute (USA)
Peter Müller, Microsoft Research (USA)
James Noble, Victoria University of Wellington (New Zealand)
Johan Oetlund, Purdue University (USA)
Alex Potanin, Victoria University of Wellington (New Zealand)
Germán Puebla, Technical University of Madrid (Spain)
Jan Smans, Katholieke Universiteit Leuven (Belgium)
Rok Strniša, University of Cambridge (UK)
Alex Summers, Imperial College (UK)
Tiphaine Turpin, IRISA/INRIA (France)
Jan Vitek, Purdue University (USA)
Stefan Wehr, University of Freiburg (Germany)
Tobias Wrigstad, Purdue University (USA)