

BOOGIE 2011

First International Workshop On Intermediate Verification Languages

<http://research.microsoft.com/~moskal/boogie2011>

colocated with
CADE 23
Wrocław, Poland

Goals

An intermediate verification language (IVL), like **Boogie** or **Why**, is used as a stepping stone between a source language and a reasoning engine. IVLs promote modularization and sharing of infrastructure. For example, the same IVL can have multiple source language front-ends and multiple reasoning engine back-ends, forming a verification tool bus. The goal of the Boogie Workshop is to advance theory and techniques supporting IVLs, to bring together researchers working with IVLs, and to promote sharing of infrastructure that they build.

The workshop is intended for topics related to any intermediate verification language, not just Boogie.

Dates

submissions: **May 1st, 2011**
notifications: June 1st, 2011
final versions: July 1st, 2011
workshop: August 1st, 2011

Program committee

Tayfun Elmas, UC Berkeley
K. Rustan M. Leino, Microsoft Research
Claude Marché, INRIA
Michał Moskal, Microsoft Research
Shaz Qadeer, Microsoft Research
Jan Smans, KU Leuven
Alexander J. Summers, ETH Zürich

Rustan and Michał are co-chairs this year.

Papers

We welcome submissions up to **12 pages in LNCS format**. The accepted papers will be printed in **informal** proceedings distributed to the participants of the workshop. With the exception of survey and history papers, the papers should contain original work which has not been submitted or accepted for publication elsewhere. Submissions instructions can be found on the workshop website.

Topics

- IVL tools
- language features of an IVL (e.g., angelic choice, tressa)
- type systems for an IVL
- translation from a source language to an IVL
- property inference at the level of an IVL (e.g., predicate abstraction, abstract interpretation, termination metric inference, Houdini-style inference)
- IVL to IVL translations (e.g., optimizations, slicing, translation between different IVLs)
- translation from an IVL to reasoning engine (SMT, ATP, HOL) input
- interaction with reasoning engines
- interpretation of reasoning engine output in terms of the source language via an IVL
- symbolic execution for an IVL
- axiomatizations of useful theories (like sets, sequences, heaps, ...)
- user experience improvements (e.g., caching of verification results)
- novel uses of IVLs (e.g., refinement or symbolic diff)
- experimental evaluations (e.g., comparing different logical encodings or reasoning engines)