10252 Executive Summary
Game Semantics and Program Verification
— Dagstuhl Seminar —

Paul-André Melliès[1], Andrzej S. Murawski[2], Andrea Schalk[3] and Igor Walukiewicz[4]

[1] University VII - Paris, FR
mellies@pps.jussieu.fr
[2] University of Oxford, GB
Andrzej.Murawski@comlab.ox.ac.uk
[3] University of Manchester, GB
A.Schalk@cs.man.ac.uk
[4] LaBRI - Bordeaux, FR
igw@labri.fr

The seminar took place from 20th until 25th June 2010. Its primary aim was to bring together researchers working on modelling programs/proofs using games and the verification community. It was clear to us that both communities could, at this point in time, begin to profit from the methods and insights gained by the other community, and be able to help with some of the other side's unsolved problems. So far the two groups have had very little interaction with each other, although there are some researchers who are active in both areas.

We organized the schedules on a day-to-day basis, in order to be as reactive as possible to the requests and questions coming from the discussions. We were also careful to leave a lot of time for interaction, while offering most participants the opportunity to give a talk. Twenty-two talks were delivered during the meeting.

1 Scientific content

The field of program verification aims to identify and implement techniques for automatic certification of program correctness or desirable program behaviour. A central task in any software verification project is the choice of a modelling approach and a decidable formalism in which the model will be represented for the purpose of verification.

Game semantics uses the metaphor of game playing to interpret computation, which it views as an exchange of moves between two players. This allows for a very concrete account of interaction consisting of sequences of moves, one that can be readily represented with common formalisms used in verification. As it turns out, this opens up the way to numerous applications. On the more abstract level, game semantics – as a modelling technique – offers a sophisticated abstraction mechanism, which enables one to describe what is observable in a program behaviour rather than what internal symbolic steps the programs make.

Our seminar began with two tutorial talks (Ghica, Murawski), which aimed to introduce the basic principles behind game semantics, outline its place in the

Dagstuhl Seminar Proceedings 10252
Game Semantics and Program Verification
http://drops.dagstuhl.de/opus/volltexte/2010/2793

landscape of programming language semantics and convey its flavour. Ghica's talk was a survey of domains in which game semantics has been applied to date, including static analysis, equivalence checking and hardware synthesis. He also described verification tools whose engine is founded on game-semantic techniques, and outlined a variety of techniques employed to guard against state explosion in game models. Murawski discussed the anatomy of game models and presented a simple model that could be described with regular expressions. The model has formed the core of several tools implemented so far and is a good point of entry into the area. The concept of strategy composition was then introduced in detail and its crucial role in constructing game models was elaborated. Finally, the principle of full abstraction was mentioned along with the advantages that it offers in verification tasks.

On the first day we also had extended talks on topics in which game semantics has already proved to be an effective tool or seems to be emerging as a potentially useful technique to ensure further progress. Kobayashi discussed model-checking functional programs via higher-order recursive schemes. The first decidability procedure for the latter was based on games and broke a long-standing stalemate in the field several years ago. Since then, alternative approaches were proposed (notably using suitably crafted type systems) and it remains to be seen what role game models will play in the future in that area. Kobayashi outlined the directions in which research on the analysis of functional programs is proceeding and suggested a few problems which game semantics might help to address. Niwiński later gave a complementary talk about an automata-theoretic approach to analyzing higher-order recursive schemes. Hofmann, in turn, described a problem (side-effect freeness) that he and his collaborators have been attacking using techniques based on logical relations. Quite interestingly, it turned out that the solution they have arrived at can be interpreted in a natural way in the spirit of game semantics, which calls for further investigation. Other unifying talks also included Seidl's perspective on the interplay of game theory and abstract interpretation, and Dal Lago's work on a compositional approach to sublinear complexity, with interesting connections to functional programming.

In the course of the week, approximately half of the talks were devoted to current topics in verification, while the other half concerned developments in game semantics. A variety of topics were covered: Habermehl lectured on regular model checking, Leroux talked about about Presburger invariants and Petri-net reachability, Sutre covered the latest results in analyzing pushdown concurrent systems, Lozes presented the latest work in verifying heap-manipulating programs and Tzevelekos talked about reachability in the functional setting.

Concurrency theory featured prominently in several talks. Melliès explained how to clarify the connections between game semantics and verification by applying ideas from concurrency theory: after recalling the tree-automata techniques applied by Ong in order to establish the decidability of mu-calculus formulas on higher-order recursion schemes, he explained how these techniques are inherently connected to the positionality property of innocent strategies in asynchronous games. On the verification side, Müller-Olm gave a survey of his work on ana-

lyzing threads and procedures, while Zhang talked about verifying probabilistic concurrent systems. Foundational aspects of model-checking partial-order models in concurrent extensions of the mu-calculus were also discussed by Gutierrez.

A number of talks were also concerned with extending the range of game semantics to new settings. Levy started with a talk describing how to derive strategies from programs via transition systems. Then, Laurent gave an overview of the literature on logic and game semantics, with a special emphasis on the notion of innocent strategies and its relationship to intuitionistic and classical logics. Zeilberger developed this direction, and explained how to think of game semantics in a purely syntactic way, using extensions of traditional sequent calculus. Then, Clairambault discussed calculating least and greatest fixed points in game models, while Laird showed his latest results on modelling polymorphism. Goyet described an extension of the usual syntax of the lambda-calculus in order to obtain a full definability result for general (not necessarily innocent) strategies on arena games.

The atmosphere during the seminar was very good, clearly all the participants were open to new ideas from 'the other side'. In particular the introductory talks attracted a number of questions asking for clarification on various issues. This showed us as the organizers that people were keen to understand the material that was presented to them, and that our selection of topics was suitable for our purposes. The periods we left unscheduled as well as the meals were then available for further discussion. In particular the young researchers present expressed their delight with the opportunity to talk to established participants in a relaxed atmosphere. Because this was a residential workshop, people did not have to worry about returning to their accommodation, or making arrangements for meals, which greatly facilitated smaller groups having additional discussions, of which we saw quite a few.

The Dagstuhl staff were extremely helpful throughout the meeting and, because most of the organizational tasks were carried out by them, the participants could concentrate on scientific matters. As the organizers we were very grateful for all the support! A number of people also commented positively on this aspect in their feedback forms.

It is perhaps too early to say how much of an impact our seminar will ultimately have. Because for many participants this was the first sustained encounter with the other community, it will take some time for ideas to be digested and adopted. The main achievement of the meeting is the creation of a platform on which new collaborations can be built in the years to come, leading to even more synergy between game semantics and verification.

Timetable

All lectures took place in Lecture Hall *Saarbrücken*.

Monday 21st June

09:00 – 09:15 *Welcome*
09:15 – 10:15 Dan Ghica: *Game semantics for program verification*
COFFEE BREAK/DISCUSSIONS
10:45 – 11:45 Andrzej Murawski: *Game semantics and automata*
LUNCH
14:15 – 15:15 Naoki Kobayashi: *Model-checking higher-order programs*
COFFEE BREAK/DISCUSSIONS
16:00 – 17:00 Martin Hofmann: *Purity of second-order functionals*

Tuesday 22nd June

09:00 – 10:00 Peter Habermehl: *Regular model checking*
10:00 – 10:30 Jérôme Leroux: *VAS reachability by Presburger inductive invariants*
COFFEE BREAK/DISCUSSIONS
11:00 – 11:45 Grégoire Sutre : *Reachability analysis for pushdown concurrent systems*
LUNCH
14:00 – 14:45 Helmut Seidl: *Strategy iteration: abstract interpretation meets game theory*
15:00 – 15:30 Nikos Tzevelekos: *Functional reachability*
COFFEE BREAK/DISCUSSIONS
16:00 – 16:45 Paul Levy: *Operational game semantics*

Wednesday 23rd June

09:00 – 10:00 Markus Müller-Olm: *Optimal analysis of threads and procedures*
COFFEE BREAK/DISCUSSIONS
10:15 – 11:45 Olivier Laurent: *Game semantics for logic*
Paul-André Melliès: : *Asynchronous games*
12:00 **Group Photo Session** (front of the chapel)
LUNCH

EXCURSION

Thursday 24th June

09:00 – 10:00 Damian Niwiński: *A gentle introduction to panic automata*
Noam Zeilberger: *Polarity and double-negation translation*
COFFEE BREAK/DISCUSSIONS
10:15 – 11:45 Julian Gutierrez: *Model-checking partial-order models of concurrency*
Lijun Zhang: *Concurrency and composition in a stochastic world*
LUNCH
14:00 – 15:30 Pierre Clairambault: *Fixed points in games*
James Laird: *Genericity and polymorphism*
COFFEE BREAK/DISCUSSIONS
16:00 – 16:45 Ugo Dal Lago: *Functional programming in sublinear space*

Friday 25th June

09:15 – 10:45 Alexis Goyet: *Lambda-bar calculus*
Etienne Lozes: *Model-checking the contracts of heap-hop*
Coffee Break/Discussions
Lunch

Participants

Pierre Clairambault (*CNRS - Paris*)
Ugo Dal Lago (*Università di Bologna*)
Ilias Garnier (*Commissariat a l'Energie Atomique - Gif-sur-Yvette*)
Dan Ghica (*University of Birmingham*)
Alexis Goyet (*CNRS - Paris*)
Charles Grellois (*ENS - Cachan*)
Julian Gutierrez (*University of Edinburgh*)
Peter Habermehl (*Université Paris Diderot*)
Martin Hofmann (*LMU München*)
Stefan Kiefer (*University of Oxford*)
Naoki Kobayashi (*Tohoku University*)
James Laird (*University of Bath*)
Olivier Laurent (*ENS - Lyon*)
Jérôme Leroux (*Université Bordeaux*)
Paul Blain Levy (*University of Birmingham*)
Etienne Lozes (*ENS - Cachan*)
Paul-André Melliès (*Université Paris Diderot*)
Markus Müller-Olm (*Universität Münster*)
Andrzej Murawski (*University of Oxford*)
Damian Niwiński (*Uniwersytet Warszawski*)
Andrea Schalk (*University of Manchester*)
Helmut Seidl (*TU München*)
Ian Stark (*University of Edinburgh*)
Grégoire Sutre (*Université Bordeaux*)
Nikos Tzevelekos (*University of Oxford*)
Paweł Urzyczyn (*Uniwersytet Warszawski*)
Noam Zeilberger (*Université Paris Diderot*)
Lijun Zhang (*Danmarks Tekniske Universitet*)