10252 Abstracts Collection
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

Abstract. From 20th to 25th June 2010, the Dagstuhl Seminar 10252 "Game Semantics and Program Verification" was held in Schloss Dagstuhl – Leibniz Center for Informatics. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Software Verification; Semantics of Programming Languages; Game Semantics; Static Analysis; Model Checking

10252 Executive Summary – Game Semantics and Program Verification

The seminar took place from 20th until 25th June 2010. Its primary aim was to foster interaction between researchers working on modelling programs/proofs using games and the verification community. The meeting brought together 28 researchers from eight different countries, both junior and senior, for a systematic assessment of what the two areas have to offer to one another, critical evaluation of what has been achieved so far, with a view to establishing common research goals for the future.

Keywords: Software Verification; Semantics of Programming Languages; Game Semantics; Static Analysis; Model Checking

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2010/2793

Dagstuhl Seminar Proceedings 10252
Game Semantics and Program Verification
http://drops.dagstuhl.de/opus/volltexte/2010/2794

Modeling induction and co-induction : from parity games to game semantics

Pierre Clairambault (CNRS - Paris, FR)

We begin the talk by a presentation of the μ-calculus. However, following the Curry-Howard correspondence, we choose to see it as a type system for inductive and co-inductive data. We define the corresponding notion of terms, inhabitants or “proofs” of μ-terms. We then recall the definition of parity games, and give the usual construction of a parity game from every term of the μ-calculus. We show that this already defines a game semantics, in the sense that (non-positional) winning strategies on the parity game generated by a μ-terms exactly correspond to inhabitants of the corresponding μ-term.

The goal of the talk is then to show how this connection can be extended to handle the implication connective, along with composition and normalization of proofs. For this purpose, we give a description of a games model mixing structures from parity games and arena game semantics. This leads to the construction of a category of win-games and winning total strategies, which is cartesian closed and has all necessary initial algebras and terminal coalgebras.

Keywords: Parity games; Game semantics; Induction; Coinduction

Functional Programming in Sublinear Space

Ugo Dal Lago (University of Bologna, IT)

We consider the problem of functional programming with data in external memory, in particular as it appears in sublinear space computation. Writing programs with sublinear space usage often requires one to use special implementation techniques for otherwise easy tasks, e.g. one cannot compose functions directly for lack of space for the intermediate result, but must instead compute and recompute small parts of the intermediate result on demand. In this paper, we study how the implementation of such techniques can be supported by functional programming languages. Our approach is based on modeling computation by interaction using the Int construction of Joyal, Street and Verity. We derive functional programming constructs from the structure obtained by applying the Int construction to a term model of a given functional language. The thus derived functional language is formulated by means of a type system inspired by Baillot and Terui's Dual Light Affine Logic. We assess its expressiveness by showing that it captures LOGSPACE.

Keywords: Lambda calculus; Game semantics; Functional programming; Implicit computational complexity

Joint work of: Dal Lago, Ugo; Schoepp, Ulrich

Full Paper:
http://springerlink.com/content/aqx8v61737338701

Game semantics for program verification

Dan Ghica (University of Birmingham, GB)

After an informal introduction of basic game-semantic concepts we examine some of the main heuristics of model checking (abstraction, CEGAR, lazy and symbolic verification, predicate abstraction) in the context of finitely-representable game models. We conclude with a look at some games-based automated software verification tools.

Keywords: Game semantics; Program verification

Full Paper:
http://doi.ieeecomputersociety.org/10.1109/LICS.2009.26

The lambda lambda-bar calculus: a calculus inspired by game semantics

Alexis Goyet (CNRS - Paris, FR)

In game semantics, the arena games present a symmetry between the player and the opponent. But the syntax of the lambda calculus does not possess this symmetry. This is due to the fact that passing and receiving arguments are fundamentally different actions: a received argument is named, and can be called at any point, multiple times; a passed argument is not named, and must be passed immediately and only once.

We mend this asymmetry by extending the lambda calculus with a “lambda-bar” construct, which names passed argument. We then allow the passing term to modify the value of an argument at any point. In particular, this allows a term to “follow” any possible interaction with its environment, and react accordingly. For example, a function may behave differently the second time that it is called.

Keywords: Lambda calculus; Lambda-bar calculus; Game semantics

Hintikka Games for Model-Checking Partial Order Models of Concurrency

Julian Gutierrez (University of Edinburgh, GB)

In this talk I will present a class of model-checking games that allows local second-order power on sets of independent transitions in the underlying partial order models where the games are played. Since the interleaving semantics of such models is not considered, some problems that may arise when using interleaving representations are avoided and new decidability results for partial order models

of concurrency are achieved. The games are shown to be sound and complete, and therefore determined. While in the interleaving case they coincide with the local model-checking games for the mu-calculus, in a partial order setting they verify properties of a number of fixpoint modal logics that can specify, in concurrent systems with partial order semantics, several properties not expressible with the mu-calculus. In particular, the games underpin a novel decision procedure for model-checking so-called temporal true-concurrency properties of a class of infinite and regular event structures.

Keywords: Modal and temporal logics; Model-checking games; Hintikka game semantics; Partial order models of concurrency

Regular Model-checking and Applications

Peter Habermehl (University Paris-Diderot, FR)

We give an overview of Regular Model-checking (RMC), a technique for verifying systems having an infinite number of configurations. The basic idea is to represent a configuration of a system as a word over a finite alphabet. Infinite sets of configurations are then conveniently represented as regular sets of words accepted by automata. Transitions of the system are given as transducers. We detail approximate reachability analysis methods based on abstraction of automata and on learning. We also give some extensions and applications of RMC.

What is a Pure Functional?

Martin Hofmann (LMU München, DE)

Given an ML function $f : (\mathsf{int} \rightarrow \mathsf{int}) \rightarrow \mathsf{int}$, how can we rigorously specify that f is pure, i.e. produces no side-effects other than those arising from calling its functional argument?

We show that existing methods based on preservation of invariants and relational parametricity are insufficient for this purpose and thus define a new notion that captures purity in the sense that, for any functional F that is pure in this sense, there exists a corresponding question-answer strategy.

This research is motivated by an attempt to prove algorithms correct that take such supposedly pure functionals as input and apply them to stateful arguments in order to inspect intensional aspects of their behaviour.

Keywords: Stateful functional programming; Logical relations; Purity; Parametricity; Fixpoint algorithms

Joint work of: Hofmann, Martin; Karbyshev, Alexandr; Seidl, Helmut

Model Checking Higher-Order Programs

Naoki Kobayashi (Tohoku University, JP)

The talk gives an overview of our recent work on higher-order program verification based on the model checking of higher-order recursion schemes.

We first show how verification problems for higher-order functional programs can be reduced to model checking problems. We then present a type-based model checking algorithm for recursion schemes, which works well in practice, despite the extremely high worst case complexity. We will also explain the limitation of the type-based algorithm, and discuss how a game-semantic view may be used for optimization of the recursion scheme model checking.

Keywords: Program verification; Higher-order recursion schemes; Model checking; Types

A game semantics of generic programming languages

James Laird (University of Bath, GB)

A fully abstract game semantics for an idealized programming language with local state and higher rank polymorphism — System F extended with general references — is described. It is quite concrete, and extends existing games models by a simple development of the existing question/answer labelling to represent “copycat links” between positive and negative occurrences of type variables, using a notion of scoping for question moves. It is effectively presentable, opening the possibility of extending existing model checking techniques to polymorphic types, for example. It is also a novel example of a model of System F with the genericity property. We prove definability of finite elements, and thus a full abstraction result, using a decomposition argument. This also establishes that terms may be approximated up to observational equivalence when instantiation is restricted to tuples of type variables.

Keywords: Genericity; Game semantics; Polymorphism; General references

See also: To appear in LICS 2010.

Game Semantics for Logic

Olivier Laurent (ENS - Lyon, FR)

Game semantics has been very successful in modelling many programming features. It is also possible to use it to provide fully complete models of various logical systems.

Starting from the innocent game model of PCF, we can obtain a fully complete model of the simply typed lambda-calculus over one atom by interpreting the atom by a two-moves game (one opponent question and one player answer), and by restricting to strategies satisfying a few conditions: forward rigidity, backward rigidity, bracketing, totality and finiteness. Thanks to the Curry-Howard correspondence, this model is fully complete for the fragment of intuitionistic logic given by formulas using one propositional variable and the implication connective. This is the two-moves model.

It is possible to show that answer moves are useless in this two-moves model of intuitionistic logic. This leads to the one-move model (where the unique atom is interpreted by the game with one move) which is also fully complete. In this model, moves correspond to occurrences of the atom in the formula, and strategies represent beta-normal eta-long forms of proofs.

Starting from these two game models, we can also build fully complete models of classical logic. The two-moves model is obtained by relaxing the bracketing condition. In the one-move case, we have to add some more structure. This is given by a notion of mu-pointer added to the usual notion of justification pointer of game semantics.

Keywords: Game semantics ; Lambda-calculus ; Intuitionistic logic ; Classical logic

The General Vector Addition System Reachability Problem by Presburger Inductive Invariants

Jérôme Leroux (Université Bordeaux, FR)

The reachability problem for Vector Addition Systems (VASs) is a central problem of net theory. The general problem is known decidable by algorithms exclusively based on the classical Kosaraju-Lambert-Mayr-Sacerdote-Tenney decomposition. This decomposition is used in this paper to prove that the Parikh images of languages accepted by VASs are semi-pseudo-linear; a class that extends the semi-linear sets, a.k.a. the sets definable in the Presburger arithmetic. We provide an application of this result; we prove that a final configuration is not reachable from an initial one if and only if there exists a Presburger formula denoting a forward inductive invariant that contains the initial configuration but not the final one. Since we can decide if a Preburger formula denotes an inductive invariant, we deduce that there exist checkable certificates of non-reachability. In particular, there exists a simple algorithm for deciding the general VAS reachability problem based on two semi-algorithms. A first one that tries to prove the reachability by enumerating finite sequences of actions and a second one that tries to prove the non-reachability by enumerating Presburger formulas.

Keywords: Presburger; Petri nets; VAS; reachability

Full Paper:
http://www.labri.fr/perso/leroux/published-papers/2009.Leroux.LICS09.pdf

Operational Game Semantics

Paul Blain Levy (University of Birmingham, GB)

In many presentations of game semantics, it is explained informally, using examples, that the strategy denoted by a term describes its behaviour. Yet this intuition is not captured in the formal definition - except for ground terms, where an adequacy result says that the operational and denotational semantics agree.

In this talk we show how to give a transition system on terms, defined using the operational semantics. Then the behaviour of each term is its trace set. The compositionality of this semantics is then a theorem, rather than a definition.

This is joint work with Soren Lassen, and closely related work has been done by Laird and by Jagadeesan, Pitcher and Riely.

Keywords: Transition systems; Traces; Compositionality

Model-checking the contracts of heap-hop

Etienne Lozes (ENS - Cachan, FR)

I will survey some recent work on the tool heap-hop, dedicated to concurrent programs synchronized by message-passing. In particular, I will introduce a new acceleration technique for queue systems that could help in analyzing a load-balancing algorithm modeled in heap-hop.

Full Paper:
http://www.lsv.ens-cachan.fr/ṽillard/heaphop/

Asynchronous games: the true concurrency of innocence

Paul-André Mèllies (University Paris-Diderot, FR)

In this survey talk, I will explain how asynchronous games offer a natural synthesis between asynchronous transition systems and arena games. In particular, I will show how to characterize/define innocent strategies by local permutation diagrams similar to local Church-Rosser diagrams in Rewriting Theory. From this local characterization follows a positionality theorem which says that innocent strategies are characterized by their halting positions. Moreover, these halting positions are precisely what the so-called relational semantics (e.g. the coherence space model) compute, this leading to a commutative triangle which provides a structural explanation for the description of beta-reduction in higher-order recursive schemes using tree automata: traversals = plays, profiles = positions, etc.

Keywords: Game semantics; True concurrency; Asynchronous transition systems; Higher-order recursive schemes

Precise Analysis of Programs with Threads and Procedures

Markus Müller-Olm (Universität Münster, DE)

Language constructs that give rise to unbounded state spaces provide a particular challenge for automatic program analysis. In my talk I presented two approaches to automatic analysis of programs with thread creation and potentially recursive procedures that we have studied in recent years. In the first part of the talk I have shown how to do gen/kill-analysis for a flow graph model of such programs. This part was based on [1]. In the second part based on [2] I sketched an automata-based approach for reachability analysis of a related program model, so-called dynamic pushdown networks, a model that extends pushdown systems by thread creation. I also sketched how iterated reachability analysis of DPNs can be used to perform live variables analysis.

[1] Peter Lammich and Markus Müller-Olm. *Precise Fixpoint-Based Analysis of Programs with Thread-Creation and Procedures.* Proceedings of CONCUR 2007, LNCS 4703.

[2] Ahmed Bouajjani, Markus Müller-Olm, and Tayssir Touili. *Regular Symbolic Analysis of Dynamic Network of Pushdown Systems.* Proceedings of CONCUR 2005, LNCS 3653.

Keywords: Automatic analysis; Threads; Procedures; Automata; Fixpoints

Game Semantics and Automata

Andrzej Murawski (University of Oxford, GB)

The talk contained an overview of game semantics as a mathematical theory for interpreting programs. I focussed on pointer games, outlined the definition of games and strategies, and described the relationship between pointers and binders in a program. Then I discussed the notion of composition of strategies and its crucial role in constructing game models. The notion of full abstraction was introduced and I mentioned several advantages of relying on it when verifying code. Finally, a concrete presentation of a simple model was given, which enables one to calculate strategies corresponding to programs of low type order.

Keywords: Game semantics; Program verification

A gentle introduction to panic automata

Damian Niwiński (University of Warsaw, PL)

The title concept generalizes automata with higher-order pushdown store by an additional option proposed by Paweł Urzyczyn and called panic.

Deterministic automata with pushdown of level n recognize (without panic) trees generated by recursion tree grammars of level n subject to the safety restriction. By adding panic we can remove this restriction. It was first showed for level 2 (by Knapik, Niwiński, Urzyczyn and Walukiewicz, 2005) and then generalized to all levels (by Hague, Murawski, Ong and Serre, 2008).

The talk has reviewed the basic concepts and illustrated them by examples, including a language suggested by Urzyczyn to distinguish between the two kinds of automata (with and without panic). Recently Paweł Parys announced the proof that the language is indeed a counterexample, i.e., it requires the panic option for automata of level 2.

Keywords: Higher-order pushdown store; Higher-order tree grammar

Strategy Iteration for Abstract Interpretation and Games

Helmut Seidl (TU München, DE)

Strategy iteration tries to approximate the least solution of a system of equations by suitable solutions of simpler systems of equations. We show how this idea can be used to construct a practical algorithm for solving systems of integer equations. We reduce solving interval equations to solving integer equations and thus strategy iteration. This approach provides us with a practical algorithm for precisely computing the abstract semantics of programs used by interval analysis of integer variables.

In the second part of the talk, we then turn to 2-player-0-sum games on finite arenas with total payoff. We show how computing the game values for these games to solving certain hierarchical systems of equations over the integers. These hierarchical systems in turn can be reduced to integer equations having a unique solution. This allows us to rely on our strategy iteration algorithm also for computing the game values of total payoff games.

Keywords: Strategy iteration; Abstract interpretation; Interval analysis; Total payoff games

Joint work of: Seidl, Helmut; Gawlitza, Thomas

Reachability Analysis of Communicating Pushdown Systems

Grégoire Sutre (Université Bordeaux, FR)

The reachability analysis of recursive programs that communicate asynchronously over reliable FIFO channels calls for restrictions to ensure decidability. We extend here a model proposed by La Torre, Madhusudan and Parlato in 2008, based on communicating pushdown systems that can dequeue with empty stack only.

Our extension adds the dual modality, which allows to dequeue with non-empty stack, and thus models interrupts for working threads. We study (possibly cyclic) network architectures under a semantic assumption on communication that ensures the decidability of reachability for finite state systems. Subsequently, we determine precisely how pushdowns can be added to this setting while preserving the decidability; in the positive case we obtain exponential time as the exact complexity bound of reachability. A second result is a generalization of the doubly exponential time algorithm of (La Torre, Madhusudan and Parlato; 2008) for bounded context analysis to our symmetric queueing policy. We present here a direct and simpler algorithm.

Keywords: Reachability analysis; Pushdown systems; Communication

Joint work of: Heussner, Alexander; Leroux, Jérôme; Muscholl, Anca; Sutre, Grégoire

Full Paper:
http://dx.doi.org/10.1007/978-3-642-12032-9_19

Functional Reachability

Nikos Tzevelekos (University of Oxford, GB)

What is reachability in higher-order functional programs? We formulate reachability as a decision problem in the setting of the prototypical functional language PCF, and show that even in the recursion-free fragment generated from a finite base type, several versions of the reachability problem are undecidable from order 4 onwards, and several other versions are reducible to each other. We characterise a version of the reachability problem in terms of a new class of tree automata introduced by Stirling at FoSSaCS 2009, called Alternating Dependency Tree Automata (ADTA). As a corollary, we prove that the ADTA non-emptiness problem is undecidable, thus resolving an open problem raised by Stirling. However, by restricting to contexts constructible from a finite set of variable names, we show that the corresponding solution set of a given instance of the reachability problem is regular. Hence the relativised reachability problem is decidable.

Keywords: Reachability analysis; Functional programs; Higher-order types

Polarity and the Logic of Delimited Continuations

Noam Zeilberger (University Paris-Diderot, FR)

Polarity in the classical sense (of linear logic, game semantics, etc.) can be seen on the one hand as a way of guiding the description of normal forms (strategies, focusing proofs, beta-normal/eta-long terms), and on the other hand as a way of decomposing the classical double-negation/continuation-passing-style translations.

From these (fairly well-accepted) views, it is natural to ask the following questions: 1. Does polarity play a role in *intuitionistic* logic?, and 2. Does polarity play a role in *delimited* continuation-passing? I propose positive answers to these questions, based on the hypothesis that (so to speak) “answers are positive”. Formally, I consider polarized logic in a more general setting where continuations are associated with positive-polarity “answer types”, thus breaking the perfect symmetry between positive and negative polarity. Purely logical considerations motivate the introduction of concepts familiar from semantics of programming languages (besides delimited continuations), notably polymorphism and monads.

Keywords: Linear logic; Classical logic; Polarized logic

Concurrency and Composition in a Stochastic World

Lijun Zhang (Technical University of Denmark, DK)

We discuss conceptional and foundational aspects of Markov automata. We place this model in the context of continuous- and discrete-time Markov chains, probabilistic automata and interactive Markov chains, and provide insight into the parallel execution of such models. We further give a detailled account of the concept of relations on distributions, and discuss how this can generalise known notions of weak simulation and bisimulation, such as to fuse sequences of internal transitions.

Keywords: Process algebra; Probabilistic automata; Interactive Markov chains; Weak bisimulation