Informal Presentations

Thursday, Sept 10

14:00–14:25	*Jesse Alama*. Expressibility of basic properties of combinatorial polyhedra in FOL and extensions
14:25–14:50	*Gilda Ferreira and Paulo Oliva*. On Bounded Functional Interpretations
14:50–15:15	*Jaime Gaspar* A logical view at Tao's finitization of principles in analysis
15:15–15:40	*David Pereira, Nelma Moreira and Simão Melo de Sousa* Encoding Kleene algebra (with tests) in Coq
15:40–16:05	*Anthony Widjaja To and Leonid Libkin* Liveness Analysis over Automatic Transition Systems

Expressibility of basic properties of combinatorial polyhedra in FOL and extensions

Jesse Alama

CENTRIA, Universidade Nova de Lisboa, Portugal alama@ftp.unl.pt

Abstract. Consider a first-order relational signature π_3 with a binary relation I and three unary predicate symbols P_0, P_1, and P_2. Intuitively, π_3 is a language for three-dimensional combinatorial polyhedra: P_0, P_1, and P_3 hold, respectively, of the 0-, 1- and 2-polytopes of a polyhedron, and I is an incidence relation for the polytopes. What properties of polyhedra can be expressed in the first-order language π_3? More generally, what properties of n-dimensional combinatorial can be expressed in an appropriate signature π_n (containing n unary predicate symbols P_0, P_1, …, P_{n-1} and a single binary relation I)? These problems, and some natural generalizations, can be solved with basic techniques of finite model theory.

How can we formally express certain properties of combinatorial polyhedra, by which we understand polyhedra considered as incidence structures (as opposed to certain kind of spatial figures or regions)?

Definition 1. *The first-order signature π_3 consists of three unary relation symbols V, E, and F, and one binary relation symbol I.*

What properties of polyhedra can be express using π_3? Can one express, for example, that a finite π_3-structure A satisfies Euler's polyhedron formula, that is, that $|V^A| - |E^A| + |F^A| = 2$? What about the property of being a homology sphere (that is, every cycle is a boundary)? What about the property that $\partial \circ \partial \equiv \emptyset$? And can we express that an π_3-structure comes from a convex three-dimensional polyhedron?

The answer to most of these questions is "no".

Theorem 1. *The properties of (1), being a homology sphere, (2) satisfying Euler's polyhedron formula, (3) satisfying $\partial_k(\partial_{k+1}(c) = \emptyset$ for all $(k+1)$-chains c, (d) being the skeleton of a convex polyhedron are all not expressible by a first-order sentence of the signature π_3.*

Some of the these properties can, however, be expressed with certain extensions of first-order logic, which we shall see.

The aforementioned properties properties are straightforwardly computable: given a finite π_3-structure A, one can obviously compute in a finite amount of time whether A satisfies Euler's polyhedron formula, whether it satisfies the property that $\partial \circ \partial \equiv \emptyset$, and whether it is the skeleton of a convex polyhedron. (The latter claim is not immediately obvious; one needs to appeal to a basic result known as Steinitz's theorem for that. Steinitz's theorem will be discussed later.) Indeed, it is clear that one can compute most of these properties in time polynomial in the cardinality $|A|$ of the structure A. Fagin's theorem [1] (which says, roughly, that existential second-order logic captures the complexity class NP) then implies that all these properties of finite π_3-structures can be captured in existential second-order logic. This investigation aims to place these properties somewhere between first-order logic and ∃-SOL.

(These questions arose from a study of the philosophy of mathematics of Imre Lakatos [2] carried out in the author's dissertation [3].)

References

1. Libkin, L.: Elements of Finite Model Theory. Texts in Theoretical Computer Science. Springer-Verlag (2004)
2. Lakatos, I.: Proofs and Refutations: The Logic of Mathematical Discovery. Cambridge University Press (1976)
3. Alama, J.: Formal Proofs and Refutations. PhD thesis, Stanford University (2009)

On Bounded Functional Interpretations

Gilda Ferreira

QMUL

A unified view over well-known interpretations of intuitionistic logic, such as Gödel's dialectica interpretation [6], Diller-Nahm interpretation [1] and Kreisel's modified realizability [7] was achieved through a parametrised interpretation in the intuitionistic logic context [8] but also, very elegantly, in the linear logic setting (see [10], [9] and [3]).

In this talk we report on work in progress concerning a general framework to the unification of the *bounded interpretations* of intuitionistic logic. This unification should include the known bounded functional interpretations whose bounds occur at the level of the interpretation of formulas, namely: bounded functional interpretation [5], bounded modified realizability [4] and confined modified realizability [2].

Similarly to the study of the interpretations that focuses in precise witnesses, in the bounded environment we also outline two different approaches towards the unification. One in the context of intuitionistic logic and the other via intuitionistic linear logic.

This is joint work with Paulo Oliva.

References

1. J. Diller and W. Nahm. Eine Variant zur Dialectica interpretation der Heyting Arithmetik endlicher Typen. *Arch. Math. Logik Grundlagenforsch*, 16:49–66, 1974.
2. G. Ferreira and P. Oliva, Confined modified realizability. To appear in *Mathematical Logic Quarterly*.
3. G. Ferreira and P. Oliva. Functional interpretations of intuitionistic linear logic. In *Proceedings of CSL 2009, LNCS 5771*, pp. 3–19. Springer, 2009.
4. F. Ferreira and A. Nunes. Bounded modified realizability. *The Journal of Symbolic Logic*, 71:329–346, 2006.
5. F. Ferreira and P. Oliva. Bounded functional interpretation. *Annals of Pure and Applied Logic*, 135:73–112, 2005.
6. K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12:280–287, 1958.
7. G. Kreisel. Interpretation of analysis by means of constructive functionals of finite types. In A. Heyting, editor, *Constructivity in Mathematics*, pages 101–128. North Holland, Amsterdam, 1959.
8. P. Oliva. Unifying Functional Interpretations. *Notre Dame Journal of Formal Logic*, 47(2):263–290, 2006.
9. P. Oliva. Modified realizability interpretation of classical linear logic. In *Proc. of the Twenty Second Annual IEEE Symposium on Logic in Computer Science LICS'07*. IEEE Press, 2007.
10. P. Oliva. Computational interpretations of classical linear logic. In *Proceedings of WoLLIC'07, LNCS 4576*, pp. 285–296. Springer, 2007.

A logical view at Tao's finitization of principles in analysis

Jaime Gaspar
(joint work with Ulrich Kohlenbach)

Fachbereich Mathematik, Technische Universität Darmstadt
Schlossgartenstrasse 7, 64289 Darmstadt, Germany
mail@jaimegaspar.com

22 June 2009

Abstract

In 2007 and 2008 Terence Tao wrote on his blog essays about the finitization of principles in analysis. His goal is to find for infinite qualitative "soft analysis" statements equivalent finitary quantitative "hard analysis" statements. These equivalences are usually proved using a contradiction and sequentially compactness argument. Tao's two prime examples are:

- a finitization of the infinite convergence principle (every bounded monotone sequence of real numbers converges);
- an almost finitization of the infinite pigeonhole principle (every colouring of the natural numbers with finitely many colours has a colour that occurs infinitely often).

We take a logical look at Tao's essays and make mainly two points:

- the finitizations can be done in a systematic way using proof theoretical tools, namely Gödel (Dialectica) functional interpretation;
- Heine-Borel compactness arguments are preferable to sequentially compactness arguments, for reverse mathematics reasons.

These points are then illustrated in a case study: the almost finitization of the infinite pigeonhole principle.

References

[1] Terence Tao, *Soft analysis, hard analysis, and the finite convergence principle.* http://terrytao.wordpress.com, 2007.

[2] Terence Tao, *The correspondence principle and finitary ergodic theory.* http://terrytao.wordpress.com, 2008.

[3] Jaime Gaspar and Ulrich Kohlenbach, *On Tao's "finitary" infinite pigeonhole principle.* To appear in *The Journal of Symbolic Logic*.

Encoding Kleene Algebra (with tests) in Coq *

Nelma Moreira[1], David Pereira[1]** and Simão Melo de Sousa[2]

[1] DCC-FC & LIACC – University of Porto
Rua do Campo Alegre 1021, 4169-007
Porto, Portugal
{nam,dpereira}@ncc.up.pt
[2] LIACC & DI – University of Beira Interior
Rua Marquês d'Ávila e Bolama
6201-001 Covilhã, Portugal
desousa@di.ubi.pt

Abstract

Kleene algebra [1], (KA) normally called *the algebra of regular events*, is an algebraic system that axiomatically captures properties of several important structures arising in Computer Science, and has been applied in several contexts like automata and formal languages, semantics and logic of programs, design and analysis of algorithms, among others. *Kleene algebra with tests* (KAT) [2] extends KA with an embedded *Boolean algebra* and is particularly suited for the formal verification of propositional programs. In particular, KAT subsumes *propositional Hoare logic* (PHL) [3], a weaker Hoare logic without the assignment axiom. This part of our formalization is described in detail by Pereira and Moreira in [4].

Here we describe a formalization of a fragment of formal languages in the Coq theorem prover. This formalization's goal is to provide a Coq library that contains proof tactics for automatically proving equivalence of KA and KAT's equational logics. Having these tactics available requires the codification of KA and KAT, and also providing proofs that they are complete for their standard models, that is, regular languages and Kozen's *automata on guarded strings* [5], respectively. In order to provide a proof that regular languages are a model of KA, we have encoded regular languages, by extending Coq's *Ensembles* library of basic set theory with new inductive types for the concatenation and Kleene's star operations, based in the work of J.C. Filliâtre [6].

In what concerns to KAT, besides the Coq modules describing KAT's signature and of proofs of its main properties, we have encoded PHL deductive rules as KAT expressions and proved that they are KAT theorems. We have also proved correct an annotated version of PHL's deductive rules in our framework.

Currently, we are implementing a decision procedure for the equivalence of KA terms, that leads to a decidable procedure for the equational theory of KA, based on the notion of Brzozowski's *derivative* [7] of a regular expression. This approach differs from the standard approach for deciding regular expression equivalence in the sense that it does no rely on comparing the minimal deterministic automata corresponding to the regular expressions being tested. We have encoded the notion of derivative of a regular expression and also proved that the derivative of a regular expression correspond to the left-quotient of the language of the original regular expression. We are currently proving that the number of derivatives of a set of regular expressions modulo ACI (associativity, commutativity and idempotence) is finite. This proof will then serve as an argument for a general recursive function that implements Brzozowski's decision procedure [8]. Since this decision procedure cannot be described by a structurally recursive function, we don't have program termination for free. In Coq, a standard solution is to use as an artificial argument that is structurally decreasing and that reflects the behaviour of the decision procedure. In particular, we are interested in using the known upper-bounds of the number of derivatives of a regular expression to be such argument. We intend to extend this procedure to KAT by using Kozen's co-algebraic approach [9], where derivatives of regular expressions were extended to KAT.

We are also particularly interested in *Schematic* KAT (SKAT) [10], a specialization of KAT involving an augmented syntax to handle first-order constructs and restricted semantic actions whose intended semantics coincides with the semantics of first-order *flowchart schemes* over a ranked alphabet Σ. SKAT programs can be transformed into KAT expressions, by converting SKAT's logical constructs into KAT Boolean tests, and converting SKAT variable assertions to KAT program symbols. In this setting, we can prove the correctness of programs using full first-order Hoare logic within our formalization, by manually converting SKAT programs into KAT expressions. We intend to automatize this task, following the lines of Aboul-Hosn and Kozen in the development of the KAT-ML [11] interactive theorem prover.

Our motivation for this work comes from the fact that we envision the usage of (an extension of) our formalization as the formal system where we can be encode and prove *proof obligations* in the context of *Design by Contract* [12] for the *Proof Carrying Code* [13] paradigm.

* This work was partially funded by Funda cão para a Ciência e Tecnologia (FCT) and program POSI, and by RESCUE project PTDC/EIA/65862/2006.
** David Pereira is funded by FCT grant SFRH/BD/33233/2007

References

1. Kleene, S. In: Representation of Events in Nerve Nets and Finite Automata. Shannon, c. and mccarthy, j. edn. Princeton University Press, Princeton, N.J. 3–42
2. Kozen, D.: Kleene algebra with tests. Transactions on Programming Languages and Systems **19**(3) (May 1997) 427–443
3. Kozen, D., Tiuryn, J.: On the completeness of propositional Hoare logic. In: RelMiCS. (2000) 195–202
4. Pereira, D., Moreira, N.: KAT and PHL in Coq. Computer Science and Information Systems **05**(02) (December 2008) ISSN: 1820-0214.
5. Kozen, D.: Automata on guarded strings and applications. Technical report, Cornell University, Ithaca, NY, USA (2001)
6. Filliâtre, J.C.: Finite Automata Theory in Coq: A constructive proof of Kleene's theorem. Research Report 97–04, LIP - ENS Lyon (February 1997)
7. Brzozowski, J.A.: Derivatives of regular expressions. JACM **11**(4) (October 1964) 481–494
8. Brzozowski, J.A.: Roots of star events. Journal of the ACM (JACM **14**(3) (Jul 1967)
9. Kozen, D.: On the coalgebraic theory of Kleene algebra with tests. Computing and information science technical reports, Cornell University (March 2008)
10. Angus, A., Kozen, D.: Kleene algebra with tests and program schematology. Technical Report TR2001-1844, Cornell University (2001)
11. Aboul-Hosn, K., Kozen, D.: KAT-ML: An interactive theorem prover for Kleene algebra with tests. Journal of Applied Non-Classical Logics **16**(1–2) (2006) 9–33
12. Meyer, B.: Applying "design by contract". Computer **25**(10) (1992) 40–51
13. Necula, G.C.: Proof-carrying code. In: POPL '97: Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, New York, NY, USA, ACM (1997) 106–119

Liveness Analysis over Automatic Transition Systems

Anthony Widjaja To and Leonid Libkin

LFCS, School of Informatics, University of Edinburgh
anthony.w.to,libkin@ed.ac.uk

Many real-world systems are more suitably represented as infinite, rather that finite-state transition systems. Some potential sources of infinity include unbounded number of processes, unbounded stacks/queues, and unbounded numeric variables. The past decade saw a lot of effort in extending the tools and techniques of model checking to handle infinite-state systems. The main hurdle one has to face in such an endeavor is that in general model checking infinite-state systems is undecidable. Broadly speaking, there are two approaches to circumvent such a problem. The first approach concerns finding subclasses of infinite systems with decidable properties of interests (e.g. safety and liveness). Such subclasses include pushdown systems, prefix-recognizable systems, and timed systems. At the other extreme, one might start with a broad class of infinite systems and develop semi-algorithms of various kinds (e.g. ones that are guaranteed to terminate but might also give a "don't know" answer).

In this talk, we briefly present some results from a conference paper [6] and some unpublished results from the PhD thesis of the first author. We consider the generic class of automatic transition systems [2] whose domain is represented by a set of words, while the transition relations are represented by (finite) synchronous transducers over words. Although model checking first-order logic over such a class is decidable (e.g. see [2]), it is known that checking safety, liveness, and, more generally, LTL-expressible properties is undecidable.

We are primarily interested in checking liveness and LTL-expressible properties. Define *recurrent reachability* over automatic transition systems to be the problem of checking whether there exists an infinite path in the given automatic transition system $\mathcal{S}$ from a given configuration s_0 (i.e. word) that visits a given regular "target" set T infinitely often. We first make an easy observation that using the classical Vardi-Wolper conversion of LTL formulae into Büchi automata [7], liveness and LTL-expressible properties over automatic transition systems can be effectively (and even quite "efficiently") reduced to the problem of recurrent reachability.

To alleviate the problem of undecidability for recurrent reachability, we then propose a semantic (i.e. not necessarily decidable) condition **(C1)** on the general class of automatic transition systems: that the transitive closure relation $\to^+$ is effectively regular and that a transducer R^+ for $\to^+$ is computable from the given input transducers. We shall later see that such a condition is not too restrictive for two reasons: 1) many decidable subclasses of infinite systems satisfy this condition, and 2) many quite successful semi-algorithms have been implemented whose goal is to compute R^+. The following was shown in [6].

Theorem 1 ([6]). *Given an automatic transition system $\mathcal{S}$ satisfying* **(C1)** *(i.e. a transducer R^+ for $\to^+$ is available as input), an input word w, and a regular set T, the problem of recurrent reachability is solvable in time $O(|R^+|^3 \times |T|^2)$. Furthermore, an NFA of size $O(|R^+|^2 \times |A|)$ recognizing the set of all w satisfying the recurrent reachability property is computable in that time bound.*

We shall emphasize now that this theorem is by no means obvious since in proving it *one has to take into account non-looping infinite paths*, i.e. infinite paths that do not visit any configurations twice. A restriction, considered in the literature, to *length-preserving transducers* (i.e., $(s, s') \in R$ implies $|s| = |s'|$) reduces recurrent reachability to reachability; however, we do *not* make this assumption, as many interesting classes of infinite-state transition systems do not satisfy it (e.g., pushdown systems, and other examples listed below). The proof of the theorem combines Ramsey theory techniques to obtain a compact representation of an infinite path with automata techniques.

We apply the above theorem to solving LTL model checking over specific classes of automatic transition systems satisfying **(C1)**. In particular, our results apply to the following classes:

- *Pushdown systems.* In this case, we derive an optimal upper bound which is exponential in the size of the LTL formula and polynomial in the size of the system. This matches the known bound of [4].
- *Prefix-recognizable systems.* In this case, we also match an optimal upper bound of [5] which is exponential in both the size of the LTL formula and the size of the system.
- *Reversal-bounded counter systems.* In this case, we derive an algorithm which is double-exponential in the size of the LTL formula (but single-exponential in the size of the specification if it is given as a Büchi automaton) and single-exponential in the size of the system and the number of counters. This upper bound on the problem is new (decidability was obtained in [3]), but it is open whether such a bound is optimal.
- *Reversal-bounded counter systems with discrete-timed clocks and one extra real counter.* In this case, we derive an algorithm which is double exponential in the size of the LTL formula and the number of clocks, but is single-exponential in the size of the system and the number of counters. Even decidability for this class of systems was open (see [3]). The upper bound is not known to be tight.

We have also obtained an initial experimental results. We have implemented a prototype of our algorithm in combination with the tool FAST [1] restricted to the generic class of counter systems with Presburger-definable transition relations. We have successfully verified a particular liveness property called *freedom from global starvation* for many cache-coherence protocols in a fully-automatic way. Most were verified in under ten minutes, the bulk of the time were spent in computing by the tool FAST [1] for computing transducers for the transitive closure relations.

References

1. S. Bardin, A. Finkel, J. Leroux, L. Petrucci. FAST: acceleration from theory to practice. *STTT* 10(5): 401–424 (2008)
2. A. Blumensath and E. Grädel. Automatic structures. In *LICS '00*, pages 51–60.
3. Z. Dang, O. Ibarra, P.S. Pietro. Liveness verification of reversal-bounded multi-counter machines with a free counter. In *FSTTCS'01*, pages 132–143.
4. J. Esparza, D. Hansel, P. Rossmanith, and S. Schwoon. Efficient algorithms for model checking pushdown systems. In *CAV '00*, pages 232–247.
5. O. Kupferman, N. Piterman, M. Vardi. Model checking linear properties of prefix-recognizable systems. In *CAV 2002*, pages 371–385.
6. A. W. To and L. Libkin. Recurrent reachability analysis in regular model checking. In *LPAR'08*, pages 198–213.
7. M. Y. Vardi, P. Wolper. Automata-theoretic techniques for modal logics of programs. *JCSS* 32(2): 183–221 (1986).