Volume 2, Number 2
April 2015

Published by the Association for Computing Machinery Special Interest Group on Logic and Computation

SIGLOG news

TABLE OF CONTENTS

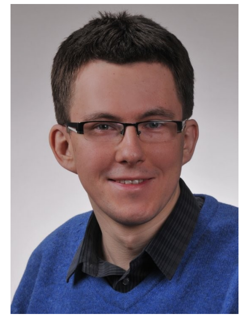
Association for Computing Machinery
Advancing Computing as a Science & Profession

SIGLOG NEWS

Published by the ACM Special Interest Group on Logic and Computation

SIGLOG News (ISSN 2372-3491) is an electronic quarterly publication by the Association for Computing Machinery.

From the Editor



Welcome to the April issue of SIGLOG News!

In this issue

– Joël Ouaknine and James Worrell survey the latest results on linear recurrence sequences in Mikołaj Bojańczyk's Automata Column.

– In Michael Mislove's column on Semantics, Alexandra Silva gives an introduction to the coalgebraic method.

– Neha Rungta's Verification Column features an article by Arie Gurfinkel, Temesghen Kahsai and Jorge A. Navas on algorithmic logic-based verification.

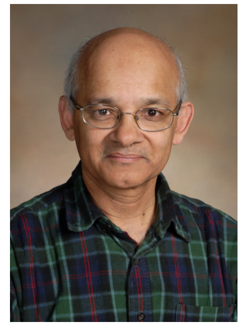
– There are many calls for papers and participation to look through in our Bulletin.

SIGLOG News is still looking for more volunteers for coordinating sections on conference reports and book reviews. Please email `editor@siglog.org` if you are interested.

Happy reading!

Andrzej Murawski
University of Warwick
SIGLOG News Editor

Chair's Letter



I am sure all of you (at least in the Northern Hemisphere) are looking forward to the end of Winter and start of the conference season. There will be a number of conferences of interest to SIGLOG members. There are two joint meetings that should produce meetings with an extra spark. This June there will be a joint meeting of CALCO and MFPS to be held in Nijmegen. Later there will be a joint meeting of LICS and ICALP in Kyoto. At the latter event there will be a special event to celebrate the 30th LICS conference.

SIGLOG is working to incorporate more meetings as "in cooperation" events with SIGLOG. I hope to be able to announce some of these soon. One of the obstacles to this is the reluctance of conferences to see their proceedings end up in the ACM Digital Library. The stumbling block is the pay wall which many view as an obstacle to accessibility. I would love to hear the opinion of the community.

We have three great columns in this issue. Enjoy!

Prakash Panangaden
McGill University
ACM SIGLOG Chair

AUTOMATA COLUMN

MIKOŁAJ BOJAŃCZYK, University of Warsaw
bojan@mimuw.edu.pl

In this automata column, Joël Ouaknine and James Worrell beautifully write about beautiful recent work on linear recurrence sequences, which are sequences of integers defined by recursion like in the Fibonacci sequence. Clearly such sequences can appear almost everywhere, so there is no particularly good reason to talk about them in the automata column, but the authors politely point out such sequences can be used to count the number of words of given length accepted by an automaton. Thanks to this connection we get to have an automata column that cites Terrence Tao and uses logarithms of algebraic numbers.

On Linear Recurrence Sequences and Loop Termination

Joël Ouaknine
Department of Computer Science
Oxford University, UK

James Worrell
Department of Computer Science
Oxford University, UK

A sequence of real numbers is said to be *positive* if all terms are positive, and *ultimately positive* if all but finitely many terms are positive. In this article we survey recent progress on long-standing open problems concerning deciding the positivity and ultimate positivity of integer linear recurrence sequences. We briefly discuss some of the many contexts in which these problems arise, and relate them to the well-known Skolem Problem, which asks whether a given linear recurrence sequence has a zero term. We also highlight some of the mathematical techniques that have been used to obtain decision procedures for these problems, pointing out obstacles to further progress.

In the second half of this survey we move on to closely related questions concerning the termination of linear while loops. This is a well-studied subject in software verification and by now there is rich toolkit of techniques to prove termination in practice. However the decidability of termination for some of the most basic types of loop remains open. Here again we discuss recent progress and remaining open problems.

1. INTRODUCTION

Linear recurrence sequences (LRS), such as the Fibonacci numbers, arise in a surprisingly diverse range of contexts, particularly in mathematics and computer science, but also in fields such as physics, biology, and economics. Connections to automata theory abound: for example, let A be a (deterministic or nondeterministic) finite automaton over some alphabet, and for every non-negative integer n, write u_n to denote the number of distinct words of length n accepted by A. Then the sequence $\langle u_0, u_1, u_2, \ldots \rangle$ of integers is an LRS. Moreover, given any integer LRS $\boldsymbol{v} = \langle v_0, v_1, v_2, \ldots \rangle$, one can always find two finite automata A and B such that each v_n is precisely the difference between the number of words of length n accepted by A and the number of words of length n accepted by B.

In this short survey, we examine connections between LRS and the termination of linear loops. The latter is a central problem in software verification that has attracted a substantial amount of attention over the last three decades, and has led to the development of tools such as Microsoft Research's TERMINATOR and T2 [Cook et al. 2011]. See also the surveys by Ben-Amram and Genaim [Ben-Amram and Genaim 2014], and by Gasarch [Gasarch 2015] describing semi-algorithmic approaches to termination based on *ranking functions*. By contrast, our interest here lies in *decidability* (and complexity) questions. Perhaps surprisingly, such questions are in turn related to deep problems and techniques from analytic and algebraic number theory, Diophantine geometry, and real algebraic geometry.

Let us start with some definitions. A (real) LRS is an infinite sequence $\boldsymbol{u} = \langle u_0, u_1, u_2, \ldots \rangle$ of real numbers having the following property: there exist real constants

$a_1, a_2, \ldots, a_k$ such that, for all $n \geq 0$,

$$u_{n+k} = a_1 u_{n+k-1} + a_2 u_{n+k-2} + \ldots + a_k u_n \,. \tag{1}$$

If the initial values $u_0, \ldots, u_{k-1}$ of the sequence are provided, the recurrence relation defines the rest of the sequence uniquely. The smallest k for which an LRS obeys such a recurrence relation is the ***order*** of the LRS.

Given an LRS $\boldsymbol{u}$ satisfying the recurrence relation (1), the ***characteristic polynomial*** of $\boldsymbol{u}$ is

$$p(x) = x^k - a_1 x^{k-1} - \ldots - a_{k-1} x - a_k \,.$$

An LRS is said to be ***simple*** if its characteristic polynomial has no repeated roots. Simple LRS, such as the Fibonacci sequence, possess a number of desirable properties which considerably simplify their analysis. They constitute a large and well-studied class of sequences, and correspond to the iterated application of *diagonalisable* matrices, as we explain shortly.

Real LRS can equivalently be characterised as those sequences $\boldsymbol{u} = \langle u_0, u_1, u_2, \ldots \rangle$ admitting an *exponential-polynomial* representation

$$u_n = \sum_{i=1}^{r} P_i(n) \rho_i^n + \sum_{i=1}^{s} Q_i(n) \lambda_i^n + \overline{Q_i(n) \lambda_i^n} \,,$$

where the ρ_i are distinct real numbers, the P_i are non-zero polynomials with real coefficients, the λ_i are distinct complex numbers, and the Q_i are non-zero polynomials with complex coefficients. Such a sequence is an LRS of order $r+2s$, with characteristic roots $\rho_1, \ldots, \rho_r, \lambda_1, \overline{\lambda_1}, \ldots, \lambda_s, \overline{\lambda_s}$. Under this representation, $\boldsymbol{u}$ is a simple LRS if and only if each of the polynomial terms is constant.

Motivated by questions in language theory and formal power series, Rozenberg, Salomaa, and Soittola [Salomaa and Soittola 1978; Rozenberg and Salomaa 1994] highlighted the following four decision problems concerning LRS over rational numbers. Given an LRS $\boldsymbol{u}$ as above:

(1) Does $u_n = 0$ for some n?
(2) Does $u_n = 0$ for infinitely many n?
(3) Is $u_n \geq 0$ for all n?
(4) Is $u_n \geq 0$ for all but finitely many n?

Problem 1 is known as the ***Skolem Problem***, after the Skolem-Mahler-Lech Theorem [Everest et al. 2003; Halava et al. 2005] which characterises the set $\{n \in \mathbb{N} : u_n = 0\}$ of zeros of an LRS $\boldsymbol{u}$ as semi-linear, i.e., consisting of a finite set together with a finite number of (infinite) arithmetic progressions. The proof of the latter is however non-effective, and the decidability of the Skolem Problem is generally considered to have been open for over 80 years [Tao 2008], a state of affairs described as "faintly outrageous" by Tao [Tao 2008] and a "mathematical embarrassment" by Lipton [Lipton and Regan 2013]. A breakthrough occurred in the mid-1980s, when Mignotte *et al.* [Mignotte et al. 1984] and Vereshchagin [Vereshchagin 1985] independently showed decidability for LRS of order 4 or less. These deep results make essential use of Baker's theorem on linear forms in logarithms (which earned Baker the Fields Medal in 1970), as well as a p-adic analogue of Baker's theorem due to van der Poorten. Unfortunately, little progress on that front has since been recorded.[1] The Skolem Problem can also be seen as a generalisation of the Orbit Problem, studied by Kannan and Lipton [Kannan and Lipton 1986, Sec. 5].

[1]A proof of decidability of the Skolem Problem for LRS of order 5 was announced in [Halava et al. 2005]. However, as pointed out in [Ouaknine and Worrell 2012], the proof seems to have a serious gap.

Interestingly and in contrast, Problem 2—hitting zero infinitely often—was shown to be decidable for arbitrary LRS by Berstel and Mignotte [Berstel and Mignotte 1976].

Problems 3 and 4 are respectively known as the ***Positivity*** and the ***Ultimate Positivity*** Problems. It is considered folklore that the decidability of Positivity (for arbitrary LRS) would entail that of the Skolem Problem [Ouaknine and Worrell 2014b], noting however that the reduction increases the order of LRS quadratically. Both Positivity and Ultimate Positivity are stated as open in literature going back at least to the 1970s (see, e.g., [Soittola 1976; Salomaa 1976; Berstel and Mignotte 1976]), as well as more recently (cf. [Halava et al. 2006; Bell and Gerhold 2007; Laohakosol and Tangsupphathawat 2009; Liu 2010; Tarasov and Vyalyi 2011; Ouaknine and Worrell 2014b], among others).

Hitherto, all decidability results for Positivity and Ultimate Positivity have been for low-order LRS; the paper [Ouaknine and Worrell 2014b] gives a detailed account of these results, obtained over a period of time stretching back some 35 years, and proves decidability of both problems for sequences of order at most 5 (with complexity in the Counting Hierarchy, itself contained within PSPACE). In addition, it is shown in [Ouaknine and Worrell 2014b] that obtaining decidability for either Positivity or Ultimate Positivity at order 6 would necessarily entail major breakthroughs in analytic number theory (more precisely regarding long-standing open problems in Diophantine approximation of transcendental numbers).

For *simple* LRS, Positivity is known to be decidable up to order 9 [Ouaknine and Worrell 2014a], and Ultimate Positivity is decidable for all orders [Ouaknine and Worrell 2014c] (in PSPACE when the order is not fixed, with a nearly matching lower bound, and in P for any fixed order—see [Ouaknine and Worrell 2014c] for details). However, beyond order 9, the algorithm for Ultimate Positivity is *non-constructive*: given an ultimately periodic LRS $\langle u_n \rangle_{n=0}^{\infty}$, the procedure of [Ouaknine and Worrell 2014c] does not produce a threshold N such that $u_n \geq 0$ for all $n \geq N$; indeed the ability to compute such a threshold would immediately yield an algorithm for the Positivity Problem for simple LRS of all orders, since the signs of $u_0, \ldots, u_{N-1}$ can be evaluated directly. In turn this would yield decidability of the Skolem Problem for simple LRS (known to be open) and also would enable one to decide the Skolem Problem for general LRS of order 5, as the only remaining unsolved sub-case involves distinct characteristic roots (cf. [Ouaknine and Worrell 2012]).

The non-constructive nature of the algorithm for deciding Ultimate Positivity of simple LRS arises from the use of lower bounds in Diophantine approximation concerning *sums of S-units*. These bounds were established in [Evertse 1984; van der Poorten and Schlickewei 1982] using Schlickewei's p-adic generalisation of Schmidt's Subspace Theorem (itself a far-reaching generalisation of the Thue-Siegel-Roth Theorem), and therein applied to study the asymptotic growth of LRS in absolute value. By contrast, [Ouaknine and Worrell 2014a] invokes Baker's Theorem on linear forms in logarithms to show decidability of Positivity for simple LRS of order at most 9. Unfortunately, while Baker's Theorem yields effective Diophantine-approximation lower bounds, it appears only to be applicable to low-order LRS. In particular, the analytic and geometric arguments that are used in [Ouaknine and Worrell 2014a] to bring Baker's Theorem to bear do not seem applicable beyond order 9.

2. EXTENDED EXAMPLE

The following extended example illustrates some of the main ideas involved in deciding positivity of an LRS, as well as some of the obstacles to generalising the decidability results described in the previous section. Full details of the method illustrated here can be found in [Ouaknine and Worrell 2014a].

Consider the sequence $\boldsymbol{u}$ given by the exponential-polynomial expression

$$u_n = \frac{33}{8} + \lambda_1^n + \overline{\lambda_1^n} + 2\lambda_2^n + 2\overline{\lambda_2^n}\,,$$

where

$$\lambda_1 = \frac{-3+4i}{5} \quad \text{and} \quad \lambda_2 = \frac{-7+24i}{25}\,.$$

This expression defines a sequence of rational numbers satisfying an order-5 linear recurrence with rational coefficients:

$$u_{n+5} = -\frac{19}{25}u_{n+4} - \frac{114}{125}u_{n+3} + \frac{114}{125}u_{n+2} + \frac{19}{25}u_{n+1} + u_n$$

whose characteristic roots are 1, λ_1, λ_2, $\overline{\lambda_1}$, and $\overline{\lambda_2}$. In this example we consider how to establish the positivity of $\boldsymbol{u}$, that is, whether $u_n \geq 0$ for all $n \in \mathbb{N}$.

Notice first that λ_1 and λ_2 both lie on the unit circle $\mathbb{T} := \{z \in \mathbb{C} : |z| = 1\}$ in the complex plane. Moreover, neither λ_1 nor λ_2 is a complex root of unity, so their respective orbits $\{\lambda_1^n : n \in \mathbb{N}\}$ and $\{\lambda_2^n : n \in \mathbb{N}\}$ are both dense in $\mathbb{T}$. In light of this one might be led to believe that u_n can be negative, e.g., if n is such that $\mathrm{Re}(\lambda_1^n)$ and $\mathrm{Re}(\lambda_2^n)$ are both at most $-\frac{3}{4}$ then clearly $u_n < 0$. Crucially, however, the joint orbit $\{(\lambda_1^n, \lambda_2^n) : n \in \mathbb{N}\}$ *is not* dense in $\mathbb{T}^2$. This is because λ_1 and λ_2 satisfy the multiplicative relationship $\lambda_1^2\lambda_2 = 1$. From this it is immediate that $\{(\lambda_1^n, \lambda_2^n) : n \in \mathbb{N}\}$ is contained in the set

$$\boldsymbol{T} := \{(z_1, z_2) \in \mathbb{T}^2 : z_1^2 z_2 = 1\}\,.$$

In fact, the set of all multiplicative relationships

$$\{(n_1, n_2) \in \mathbb{Z}^2 : \lambda_1^{n_1}\lambda_2^{n_2} = 1\}$$

among λ_1 and λ_2 is a rank-1 subgroup of $\mathbb{Z}^2$ with generator $(2, 1)$. It follows from Kronecker's theorem on simultaneous inhomogeneous Diophantine approximation [Cassels 1965] that $\{(\lambda_1^n, \lambda_2^n) : n \in \mathbb{N}\}$ is a dense subset of $\boldsymbol{T}$.

Now consider the function $f : \mathbb{T}^2 \to \mathbb{R}$ given by

$$f(z_1, z_2) = \frac{33}{8} + z_1 + \overline{z_1} + 2z_2 + 2\overline{z_2}\,.$$

Then $u_n = f(\lambda_1^n, \lambda_2^n)$ and (by calculus) f is non-negative on the set $\boldsymbol{T}$. It follows that $u_n \geq 0$ for all n.[2]

Next we extend the example by considering the order-6 LRS $\boldsymbol{v} = \langle v_0, v_1, v_2, \ldots\rangle$, given by $v_n := u_n - \frac{1}{2^n}$. We have already established positivity of $\boldsymbol{u}$. Positivity of $\boldsymbol{v}$ amounts to establishing the lower bound $\frac{1}{2^n} \leq u_n$ for all $n \in \mathbb{N}$. But this is a delicate question because u_n comes arbitrarily close to 0 as n ranges over $\mathbb{N}$. Indeed the function f above has two zeros on the set $\boldsymbol{T}$—at the point

$$(z_1^*, z_2^*) := \left(-\tfrac{1}{8} + \tfrac{\sqrt{63}}{8}i, -\tfrac{31}{32} + \tfrac{\sqrt{63}}{32}i\right)$$

and its complex conjugate. Since $\{(\lambda_1^n, \lambda_2^n) : n \in \mathbb{N}\}$ is a dense subset of $\boldsymbol{T}$ it follows that $\liminf_n u_n = \liminf_n f(\lambda_1^n, \lambda_2^n) = 0$.

[2]Note that density of $\{(\lambda_1^n, \lambda_2^n) : n \in \mathbb{N}\}$ in $\boldsymbol{T}$ is not needed to certify the positivity of $\boldsymbol{u}$, only inclusion. However the fact that the topological closure of the orbit $\{(\lambda_1^n, \lambda_2^n) : n \in \mathbb{N}\}$ is an algebraic subset of $\mathbb{C}^2$ is an important property that is relevant to the completeness of method of proving positivity illustrated in this example.

One can obtain a lower bound for $f(\lambda_1^n, \lambda_2^n)$ as a function of n via lower bounds on the distance between $(\lambda_1^n, \lambda_2^n)$ and the two zeros of f in $\boldsymbol{T}$. Baker's Theorem on linear forms in logarithms of algebraic numbers is instrumental in obtaining such bounds.

A simple form of Baker's Theorem (restricted to linear forms in three logarithms) is as follows. Let $\alpha_1, \alpha_2, \alpha_3$ be algebraic numbers and b_1, b_2, b_3 integers of absolute value at most H. Then for the principal branch $\log$ of the complex logarithm function,

$$\Lambda := b_1 \log \alpha_1 + b_2 \log \alpha_2 + b_3 \log \alpha_3$$

is either 0 or satisfies $|\Lambda| > H^{-C}$, where C is an effectively computable constant depending only on α_1, α_2, and α_3.

With this result in hand, we can bound the distance between λ_1^n and z_1^* as follows. Note that for any $\alpha \in \mathbb{T}$, we have $\log \alpha = i \arg \alpha$. Next, let n be such that $\lambda_1^n \neq z_1^*$ and choose $m \in \mathbb{Z}$ such that $-\pi < n \arg \lambda_1 - 2\pi m \leq \pi$. Then

$$\begin{aligned} |\lambda_1^n - z_1^*| \;&\geq\; \frac{1}{2}\, |n \arg \lambda_1 - 2\pi m - \arg z_1^*| \\ &\geq\; \frac{1}{\mathrm{poly}(n)}\,, \end{aligned} \tag{2}$$

where $\mathrm{poly}(n)$ is an effectively computable polynomial depending only on λ_1 and z_1^*. Here the first inequality involves estimating the distance between λ_1^n and z_1^* in terms of the length of the circular segment of $\mathbb{T}$ lying between λ_1^n and z_1^*. The second inequality follows from Baker's Theorem, using the facts that $\pi = \arg(-1)$ and $|m|$ is less than n.

We can now compute $N \in \mathbb{N}$ such that $u_n \geq \frac{1}{2^n}$ for all $n \geq N$. First, invoking ideas similar to those used by Kannan and Lipton in their work on the Orbit Problem [Kannan and Lipton 1986], one can choose N sufficiently large such that $(\lambda_1^n, \lambda_2^n)$ is not equal to either or the two zeros of the function f on $\boldsymbol{T}$ for $n \geq N$. One can then apply inequalities of type (2) to get lower bounds for the distance between $(\lambda_1^n, \lambda_2^n)$ and the zeros of f, and hence obtain a lower bound on $u_n = f(\lambda_1^n, \lambda_2^n)$ of the form $\frac{1}{\mathrm{poly}(n)}$ for some fixed effectively computable polynomial $\mathrm{poly}(n)$. Choosing N such that $\frac{1}{\mathrm{poly}(n)} \geq \frac{1}{2^n}$ for all $n \geq N$, we have $u_n \geq \frac{1}{2^n}$ for all $n \geq N$. It immediately follows that $v_n < 0$ can only happen if $n < N$. Thus the positivity of $\boldsymbol{v}$ can be established by checking positivity of v_n for $n < N$ by exhaustive search.

We conclude the example with several remarks. First, recall that an important step in establishing the positivity of $\boldsymbol{u}$ was to show that the function f only assumes non-negative values on $\boldsymbol{T}$. In the case at hand, this fact could be shown using elementary calculus. More generally one can phrase such a question as the truth of a universal sentence in the theory of real-closed fields. This is the approach taken in [Ouaknine and Worrell 2014c], which gives a procedure to decide the ultimate positivity of simple linear recurrence sequences. In the other direction, we have also shown in [Ouaknine and Worrell 2014c] that deciding the truth of such universal sentences is reducible in polynomial time to deciding the ultimate positivity of simple linear recurrence sequences.

Next, observe that our analysis of the positivity of $\boldsymbol{v}$ crucially depends on computing an *effective* sub-exponential lower bound on u_n. We were able to do this thanks to the effective nature of the constants in Baker's Theorem. In turn, our application of Baker's Theorem hinged on the fact that the function f had only isolated zeros on $\boldsymbol{T}$. This isolated-zero condition can be shown to fail for higher-order recurrences. In particular, as we have described in the previous section, our result that ultimate positivity is decidable for simple LRS of arbitrary order relies on non-effective Diophantine-approximation lower bounds in place of Baker's Theorem [Ouaknine and

Worrell 2014c]. Such results suffice for considering *ultimate* positivity as well as for handling termination of linear while loops, as we will see in the next section.

Finally, observe that the sequence $\boldsymbol{v}$ in the above example is simple. Even at order 6, if one admits linear terms in exponential polynomials then to decide positivity one provably needs much sharper Diophantine-approximation estimates for logarithms of algebraic numbers than are currently known; see [Ouaknine and Worrell 2014b] for details.

3. TERMINATION OF LINEAR LOOPS

Termination is a fundamental decision problem in program verification. In particular, termination of programs with linear assignments and guards has been extensively studied over the last decade; as discussed earlier, this has led to the development of powerful semi-algorithms to prove termination via synthesis of ranking functions, several of which have been implemented in software-verification tools.

In this survey, we focus on ***linear loops***, i.e., programs of the form:

$$\mathsf{P}: \ \boldsymbol{x} \leftarrow \boldsymbol{t}\,; \mathit{while}\ B\boldsymbol{x} \geq \boldsymbol{c}\ \mathit{do}\ \boldsymbol{x} \leftarrow A\boldsymbol{x} + \boldsymbol{a}\,,$$

where $\boldsymbol{x}$ is vector of variables, $\boldsymbol{t}$ is a vector of integer, rational, or real numbers, $\boldsymbol{a}$ and $\boldsymbol{c}$ are integer vectors, and A and B are integer matrices of the appropriate dimensions. Here the loop guard is a conjunction of linear inequalities and the loop body consists of a simultaneous affine assignment to $\boldsymbol{x}$. When vectors $\boldsymbol{a}$ and $\boldsymbol{c}$ are both zero, the loop is said to be ***homogeneous***.

The ***dimension*** of a linear loop is that of vector $\boldsymbol{x}$. For P of dimension d, we say that P ***terminates*** on a set $S \subseteq \mathbb{R}^d$ if it terminates for all initial vectors $\boldsymbol{t} \in S$. In 2004, Tiwari gave a procedure to decide whether a given linear loop terminates on $\mathbb{R}^d$ [Tiwari 2004], and two years later Braverman showed decidability of termination on $\mathbb{Q}^d$ [Braverman 2006]. However the most natural problems from the point of view of program verification are (i) termination on $\mathbb{Z}^d$ and (ii) termination on given integer singleton sets (corresponding to fixed initial starting conditions, and also known as the *Halting Problem* for linear loops).

While termination on $\mathbb{Z}^d$ reduces to termination on $\mathbb{Q}^d$ in the homogeneous case (by a straightforward scaling argument), termination on $\mathbb{Z}^d$ in the general case is stated as an open problem in [Ben-Amram et al. 2012; Braverman 2006; Tiwari 2004]. Recently, we established decidability of termination on $\mathbb{Z}^d$ for linear loops provided either that such loops have dimension at most 4 or that the update matrix A is diagonalisable [Ouaknine et al. 2015]. (The general case however remains open.)

Two observations are in order. First—recalling that the guard of linear loops consists of a conjunction of linear inequalities—a given linear loop eventually halts if and only if one of the linear inequalities is eventually violated. In the study of linear loop termination, it is therefore sufficient to restrict one's attention to loops with a single inequality as guard. Second, given an inhomogeneous linear loop P of dimension d as above, one can readily manufacture a *homogeneous* linear loop P' such that P halts on a given initial starting vector $\boldsymbol{t} \in \mathbb{R}^d$ if and only if P' halts on $\boldsymbol{t}, 1 \in \mathbb{R}^{d+1}$ (i.e., the vector $\boldsymbol{t}$ augmented by a $(d+1)$th entry of 1. When studying termination on singleton sets, we may therefore assume homogeneity at the cost of increasing the dimension by 1. Note however that the attendant transformation may fail to preserve certain properties of the update matrix, such as diagonalisability.

Let us therefore examine the following d-dimensional homogeneous linear loop:

$$\mathsf{Q}: \ \boldsymbol{x} \leftarrow \boldsymbol{t}\,; \mathit{while}\ \boldsymbol{b}^T\boldsymbol{x} \geq 0\ \mathit{do}\ \boldsymbol{x} \leftarrow A\boldsymbol{x}\,,$$

where $\boldsymbol{b}^T$ is a row vector. Writing $u_n = \boldsymbol{b}^T A^n \boldsymbol{t}$, easily follows from the Cayley-Hamilton theorem that $\boldsymbol{u} = \langle u_0, u_1, u_2, \ldots \rangle$ is an LRS of order at most d. (Conversely, any LRS of order d may be realised as such a d-dimensional linear loop.) One can moreover show that $\boldsymbol{u}$ is simple if and only if A is diagonalisable.

From the above observations and our earlier results on LRS, we immediately obtain the following: it is decidable, for any fixed starting vector $\boldsymbol{t} \in \mathbb{Z}^d$, whether Q terminates provided either that d is at most 5, or that A is diagonalisable and d is at most 9. Moreover, by way of hardness, decidability of termination in dimension 6 or above (in the non-diagonalisable case) would necessarily entail major breakthroughs in analytic number theory. Turning to the inhomogeneous program P, one can decide termination on a singleton set provided that P has dimension 4 or less, simply by homogenising.

Let us return to the question of the termination of P on $\mathbb{Z}^d$, under the assumption that A is diagonalisable. This problem can equivalently be posed in terms of whether the set NT of initial values $\boldsymbol{t} \in \mathbb{R}^d$ on which P is non-terminating contains an integer point. The set NT is easily seen to be convex. In the following example it also happens to be a *semi-algebraic* subset of $\mathbb{R}^d$, i.e., defined by a conjunction of polynomial inequalities with integer coefficients.

Example 3.1. Let θ be a fixed real number that is not a rational multiple of π, and consider the program

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \leftarrow \boldsymbol{t}\,;\, \textit{while}\ z - y \geq 0\ \textit{do}\ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \leftarrow \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

The update matrix in the loop body is a counter-clockwise rotation around the z-axis by angle θ. The set of non-terminating points is thus the cone $\{(x, y, z) \in \mathbb{R}^3 : z^2 \geq x^2 + y^2\}$.

Note that the update matrix in Example 3.1 is both diagonalisable and orthogonal. It follows that it has a basis of eigenvectors, all corresponding to eigenvalues of modulus one. More generally, consider an inhomogeneous linear loop with a diagonalisable update matrix A. Writing $\mathbb{R}^d$ as a direct sum $\mathbb{R}^d = V_1 \oplus \ldots \oplus V_m$, where each V_i is a sum of eigenspaces of A corresponding to eigenvalues of a given fixed modulus, it is shown in [Ouaknine et al. 2015] that for each i the set $\mathsf{NT} \cap V_i$ is an (effectively computable) semi-algebraic subset of $\mathbb{R}^d$. A key step to proving this result is to show how to compute all integer linear relationships among the phases of the eigenvalues associated to each V_i. Such a computation is possible thanks to number-theoretic results of Masser [Masser 1988].

While we know that $\mathsf{NT} \cap V_i$ is semi-algebraic for $i = 1, \ldots, m$, whether the set NT itself is semi-algebraic appears to be a much harder question. However for the purposes of deciding termination, this issue can be side-stepped by focusing on *eventual non-termination* rather than non-termination. We say that P is eventually non-terminating on $\boldsymbol{t} \in \mathbb{R}^d$ if, starting from initial value $\boldsymbol{t}$, after executing the loop body $\boldsymbol{x} \leftarrow A\boldsymbol{x} + \boldsymbol{a}$ a finite number of times *whilst disregarding the loop guard*, we eventually reach a value on which P fails to terminate. We write ENT for the set of eventually non-terminating initial vectors. Clearly and $\mathsf{NT} \subseteq \mathsf{ENT}$ and P is non-terminating if and only if ENT contains an integer point.

Decidability of termination in arbitrary dimension (assuming a diagonalisable update matrix) was shown in [Ouaknine et al. 2015] through an analysis of the set ENT. Given a linear loop, it is shown in [Ouaknine et al. 2015] how to compute a convex semi-algebraic set $W \subseteq \mathbb{R}^d$ such that the integer points $\boldsymbol{t} \in W$ are precisely the eventually non-terminating integer initial values. Since, by a result of Khachiyan and Porkolab [Khachiyan and Porkolab 1997], it is decidable whether a convex semi-algebraic set contains an integer point [Khachiyan and Porkolab 1997],[3] one can decide whether a linear loop is terminating on $\mathbb{Z}^d$. The computation of W makes critical use of deep number-theoretic tools such as the S-units theorem of Evertse, van der Poorten, and Schlickewei [Evertse 1984; van der Poorten and Schlickewei 1982], which as mentioned earlier played a key role in establishing decidability of Ultimate Positivity for simple LRS of all orders. Roughly speaking, these tools are used to prove that the termination of P on a given initial vector $\boldsymbol{t}$ is determined by the components of $\boldsymbol{t}$ on eigenspaces corresponding to eigenvalues of maximum modulus. Critically the non-effectiveness of these results is not a problem when considering eventual non-termination.

The set W is used in [Ouaknine et al. 2015] in lieu of the set ENT of eventually non-terminating points. It is immediate from the definition of W that W and ENT have the same topological closure, and thus [Ouaknine et al. 2015] shows *inter alia* that the topological closure of ENT is semi-algebraic. Whether ENT or NT are semi-algebraic is, to the best of our knowledge, open. We refer the reader to [Ouaknine et al. 2015] for full details.

With regard to complexity, termination of linear loops over the set of all integer points is easily seen to be coNP-hard, by reduction from integer programming, taking the update matrix A to be the identity. The procedure proposed in [Ouaknine et al. 2015], on the other hand, requires exponential space. In contrast, even though not stated explicitly in [Tiwari 2004] and [Braverman 2006], deciding termination on $\mathbb{R}^d$ and $\mathbb{Q}^d$ (which relies mainly on spectral techniques and linear algebra) can be done in polynomial time.[4]

4. OPEN PROBLEMS

Decidability of termination of linear loops remains open if we do not assume a diagonalisable update matrix. Decidability of termination is also open for the more general class of ***linear constraint loops***, that is, loops of the form

$$\boldsymbol{x} \leftarrow \boldsymbol{t}\,; \textit{while } B\boldsymbol{x} \geq \boldsymbol{c} \textit{ do } A\begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{x}'\end{pmatrix} \leq \boldsymbol{d}\ ,$$

in which the loop body consists of a conjunction of linear constraints among the "before" and "after" values of the program variables, respectively denoted $\boldsymbol{x}$ and $\boldsymbol{x}'$.

A special case of linear constraint loops feature ***octagonal constraints***, which have the form $x - y' \sim k$ or $x + y' \sim k$ for (possibly identical) program variables x and y, integer k, and comparison operator $\sim \in \{\leq, \geq\}$. It is shown in [Bozga et al. 2012] that for linear constraint loops with exclusively octagonal constraints, the set of integer points in NT is effectively semi-linear. In fact more is true: for such loops the reflexive transitive closure of the transition relation denoted by the loop body is an effectively computable semi-linear set.

The outstanding open problem in the area is the decidability of Skolem's Problem. At we have remarked above, currently decidability is only known for LRS of order at most 4. While decidability at order 2 is elementary, the proofs of decidability at orders 3 and 4 make use of Baker's Theorem. On the other hand, unlike for the Positivity Problem, it is not clear that a solution to Skolem's Problem necessarily entails progress in understanding Diophantine approximation for logarithms of algebraic numbers. In terms

[3]By contrast, recall that the existence of an integer point in an *arbitrary* (i.e., not necessarily convex) semi-algebraic set—which is equivalent to Hilbert's tenth problem—is well-known to be undecidable.
[4]This observation relies on the facts that one can compute Jordan canonical forms of integer matrices and solve instances of linear programming problems with algebraic numbers in polynomial time [Cai 1994].

of computational complexity, the current best lower bound is NP-hardness (which already holds for the restricted case of matrices with all entries either 0 or 1).

REFERENCES

J. P. Bell and S. Gerhold. 2007. On the Positivity Set of a Linear Recurrence. *Israel Jour. Math.* 57 (2007).

A. M. Ben-Amram and S. Genaim. 2014. Ranking Functions for Linear-Constraint Loops. *J. ACM* 61, 4 (2014).

A. M. Ben-Amram, S. Genaim, and A. N. Masud. 2012. On the Termination of Integer Loops. *ACM Trans. Program. Lang. Syst.* 34, 4 (2012).

J. Berstel and M. Mignotte. 1976. Deux propriétés décidables des suites récurrentes linéaires. *Bull. Soc. Math. France* 104 (1976).

M. Bozga, R. Iosif, and F. Konecný. 2012. Deciding conditional termination. In *Proc. TACAS (LNCS)*, Vol. 7214. Springer.

M. Braverman. 2006. Termination of integer linear programs. In *Proc. CAV (LNCS)*, Vol. 4144. Springer.

J.-Y. Cai. 1994. Computing Jordan Normal Forms Exactly for Commuting Matrices in Polynomial Time. *Int. J. Found. Comput. Sci.* 5, 3/4 (1994), 293–302.

J.W.S. Cassels. 1965. *An introduction to Diophantine approximation*. Camb. Univ. Press.

B. Cook, A. Podelski, and A. Rybalchenko. 2011. Proving Program Termination. *Commun. ACM* 54, 5 (2011).

G. Everest, A. van der Poorten, I. Shparlinski, and T. Ward. 2003. *Recurrence Sequences*. American Mathematical Society.

J.-H. Evertse. 1984. On sums of S-units and linear recurrences. *Compositio Mathematica* 53, 2 (1984).

W. I. Gasarch. 2015. Proving Programs Terminate Using Well-Founded Orderings, Ramsey's Theorem, and Matrices. *Advances in Computers* 97 (2015), 147–200.

V. Halava, T. Harju, and M. Hirvensalo. 2006. Positivity of second order linear recurrent sequences. *Discrete Applied Mathematics* 154, 3 (2006).

V. Halava, T. Harju, M. Hirvensalo, and J. Karhumäki. 2005. *Skolem's Problem – On the Border between Decidability and Undecidability*. Technical Report 683. Turku Centre for Computer Science.

R. Kannan and R. J. Lipton. 1986. Polynomial-Time Algorithm for the Orbit Problem. *J. ACM* 33, 4 (1986).

L. Khachiyan and L. Porkolab. 1997. Computing Integral Points in Convex Semi-algebraic Sets. In *Proc. FOCS*. 162–171.

V. Laohakosol and P. Tangsupphathawat. 2009. Positivity of third order linear recurrence sequences. *Discrete Applied Mathematics* 157, 15 (2009).

R. J. Lipton and K. W. Regan. 2013. *People, Problems, and Proofs - Essays from Gödel's Lost Letter: 2010*. Springer.

L. Liu. 2010. Positivity of Three-Term Recurrence Sequences. *Elec. J. Comb.* 17, 1 (2010).

D. W. Masser. 1988. Linear Relations on Algebraic Groups. In *New Advances in Transcendence Theory*. Camb. Univ. Press.

M. Mignotte, T. Shorey, and R. Tijdeman. 1984. The distance between terms of an algebraic recurrence sequence. *J. für die reine und angewandte Math.* 349 (1984).

J. Ouaknine, J. Sousa Pinto, and J. Worrell. 2015. On Termination of Integer Linear Loops. In *Proc. SODA*. ACM-SIAM.

J. Ouaknine and J. Worrell. 2012. Decision Problems for Linear Recurrence Sequences. In *Proc. RP (LNCS)*, Vol. 7550. Springer.

J. Ouaknine and J. Worrell. 2014a. On the Positivity Problem for Simple Linear Recurrence Sequences. In *Proc. ICALP (LNCS)*, Vol. 8573. Springer.

J. Ouaknine and J. Worrell. 2014b. Positivity Problems for Low-Order Linear Recurrence Sequences. In *Proc. SODA*. ACM-SIAM.

J. Ouaknine and J. Worrell. 2014c. Ultimate Positivity is Decidable for Simple Linear Recurrence Sequences. In *Proc. ICALP (LNCS)*, Vol. 8573. Springer.

G. Rozenberg and A. Salomaa. 1994. *Cornerstones of Undecidability*. Prentice Hall.

A. Salomaa. 1976. Growth Functions of Lindenmayer Systems: Some New Approaches. In *Automata, Languages, Development*, A. Lindenmayer and G. Rozenberg (Eds.). North-Holland.

A. Salomaa and M. Soittola. 1978. *Automata-theoretic aspects of formal power series.* Springer.

M. Soittola. 1976. On D0L Synthesis Problem. In *Automata, Languages, Development*, A. Lindenmayer and G. Rozenberg (Eds.). North-Holland.

T. Tao. 2008. *Structure and randomness: pages from year one of a mathematical blog*. American Mathematical Society.

S. P. Tarasov and M. N. Vyalyi. 2011. Orbits of Linear Maps and Regular Languages. In *Proc. CSR (LNCS)*, Vol. 6651. Springer.

A. Tiwari. 2004. Termination of Linear Programs. In *Proc. CAV (LNCS)*, Vol. 3114. Springer.

A.J. van der Poorten and H.P. Schlickewei. 1982. The growth conditions for recurrence sequences. *Macquarie Math. Reports* 82-0041 (1982).

N. K. Vereshchagin. 1985. The problem of appearance of a zero in a linear recurrence sequence (in Russian). *Mat. Zametki* 38, 2 (1985).

SEM

Semantics Column

MICHAEL MISLOVE, Tulane University
mwm@math.tulane.edu

This quarter's column is by ALEXANDRA SILVA, who works on coalgebras, among other things. Alexandra provides us with an introduction to coalgebras and their role in computational models, and uses them to analyze a fascinating illustrative example: Brzozowski's Algorithm, which takes as input a finite state automaton, and as if by magic, produces the minimal automaton accepting the same language. Unraveling this algorithm and revealing its underlying mechanism is a lovely example of coalgebras at work, and indeed, the results that Alexandra describes have been the topic of an invited lecture at LICS [Panangaden 2013].

Alexandra's presentation is a model of clarity, so there isn't much to add in terms of explaining the column itself. But what I would like to offer is some history of coalgebras, and how they came to prominence in theoretical computer science. The story starts with the Foundations of Mathematics, and Peter Aczel's remarkable insight that there is a common theme to the Milner-Park notion of bisimulation, and an alternative to the Foundation Axiom in Zermelo-Frankel Set Theory (ZF). The Foundation Axiom is the one that implies that in ZF, every set is in bijective equivalence with some set in the von Neumann hierarchy, $\mathcal{H} \equiv \bigcup_\alpha \mathcal{P}^\alpha(\emptyset)$. Another way to think of it is that the Foundation Axiom prohibits infinite descending membership chains in ZF: $A_1 \ni A_2 \ni \cdots \ni A_n \ni \cdots$; in particular, there cannot be a set $A \ni A$ that is a member of itself. Such sets are called *non-well founded*, because they violate well-foundedness property of set membership. The Foundation Axiom strikes even set theory experts as bizarre, and simply not of the same character as the other ZF axioms. A number of attempts have been made to replace the Foundation Axiom with a more natural alternative that would allow sets that are members of themselves. Researchers such as Scott [Scott 1960], Boffa [Boffa 1968] and Honsell and Forti [Forti and Honsell 1983] (an approach that is very close to Aczel's) all proposed alternatives. But Aczel's approach has proved to be the most successful of the attempts, and it underlies the growth of coalgebras in computer science. Here's how that happened.

Aczel's approach is based on his *Anti-Foundation Axiom* [Aczel 1998], where sets are represented as accessible pointed graphs (APGs), with each node denoting a set, and an edge running from a set to each of its elements. (A graph with a point is *accessible* if each node lies on some path from the point node.) This allows sets to be members of themselves, and so the usual method of validating which sets are the same – verifying the elements of each are elements of the other – just doesn't work when the set being tested is one of its own members. Aczel realized that the Milner-Park notion of bisimulation could be used for this purpose. Viewing sets *qua* APGs as automata with one action, where each edge represents reverse set membership, implies that two sets are the same iff there is a bisimulation between them. This is a coinductive definition of set membership. Coalgebras enter the picture when one looks for a model of the system in which equivalent APGs are identified. Such a model in the category of classes is the final coalgebra for the category: two APGs represent the same set precisely when their images in the final coalgebra are the same; the existence of such a final coalgebra is proved in [Aczel and Mendler 1989].

An advantage of Aczel's approach is that it allows systems of equations to be solved *up to equality*. For example the familiar presentation of a *stream* of data as a system $S = (a, S')$ where a is one datum, and S' is again a stream can be solved in ZF only up to isomorphism, but in Aczel's universe it can be solved up to equality: given an underling set A of data, there is a set $S = A \times S$ of (infinite) streams over A. An early application of these ideas was a new resolution of the Liar Paradox: *This sentence is false* [Barwise and Etchemendy 1987]. One of the computational motivations for Aczel's work was to give an alternative model for recursion, in particular in the context of process calculi such as CSP, and this led the author and two colleagues to explore the relationship between Aczel's world and domain theory [Mislove, Moss and Oles 1991]. But it fell to Jan Rutten [Rutten 1992] to develop the first denotational semantics of a simple process algebra using Aczel's theory. Jan continued his investigations into coalgebras and their application in computer science, and together with Bart Jacobs, the two founded a veritable cottage industry in the area that has flourished for many years now, as Alexandra's comments in the last section of her column amply demonstrate. In fact, Alexandra's column also shows us there are fascinating applications and insights that coalgebras offer into computational phenomena, as well as suggesting some intriguing potential applications that are still to be explored.

REFERENCES

P. Aczel. 1998 Non-well-founded Sets. *CSLI Lecture Notes 14*, Stanford, CA: Stanford University, Center for the Study of Language and Information, pp. xx+137.

P. Aczel and M. Mendler. 1989 A final coalgebra theorem. *Lecture Notes in Computer Science* 389 (1989), pp 357-365.

J. Barwise and J. Etchemendy. 1987 The Liar. Oxford University Press (1987).

M. Boffa. 1968 Les ensembles extraordinaires. *Bulletin de la Societe Mathematique de Belgique* XX:3-15 (1968).

M. Forti and F. Honsell. 1983 Set theory with free construction principles. *Ann. Scuola Norm. Sup. Pisa, Cl. Sci.* 10 (1983), pp. 493522.

M. Mislove , L. S. Moss, and F. J. Oles. 1991 Non-well-founded sets as ideal fixed points. *Information and Computation* 93 (1991), pp. 1654.

P. Panangaden. 2013 Duality in Logic and Computation. *Logic in Computer Science Symposium*, plenary lecture, June 2013.

J. Rutten. 1992 Processes as terms: non-well-founded models for bisimulation. *Mathematical Structures in Computer Science* 2 (1992), pp. 257–275.

D. Scott. 1960 A different kind of model for set theory. Unpublished paper, talk given at the *1960 Stanford Congress of Logic, Methodology and Philosophy of Science*.

A short introduction to the coalgebraic method

Alexandra Silva, Radboud University Nijmegen

This article contains a brief introduction to coalgebra and an overview of a recent proof of correctness for Brzozowski's algorithm which uses coalgebraic techniques. In the discussion section we briefly discuss the most active research lines in the coalgebra community and take a personal outlook at what future might bring for the field.

1. INTRODUCTION

Coalgebra was established in the last decades [Barwise and Moss 1996; Jacobs et al. 1997; Rutten 2000] as a mathematical framework to study dynamical systems from a modular perspective. In a nutshell, the idea behind the coalgebraic framework is that many data structures and automata-like models can be described in a uniform way as instances of a certain abstract mathematical concept. To illustrate in more detail what this means, consider possibly infinite lists over elements of a given set and deterministic finite automata (DFA).

The type of potentially infinite lists is built-in in many functional languages. For instance, in OCaml it is represented by `'a list` and can be instantiated for two cases, namely `[]` for the empty list and `hd :: tl` for the list with head `hd` of type `'a` and tail `tl` of the same type `'a list`. Concrete infinite lists can then be defined using the `let rec` construct:

```
let rec ones = 1 :: ones
let rec alt  = 1 :: 2 :: alt
```

The first example defines the infinite list of ones $1, 1, 1, 1, \ldots$ and the second the alternating sequence $1, 2, 1, 2, \ldots$.

A DFA (over an alphabet A) consists of a set of states S, together with a distinguished initial state and a subset of final states, and a transition function assigning to each state $s \in S$ and letter $a \in A$ a next state $t \in S$.

At first DFA and infinite lists appear to bear little resemblance to one another: for instance, automata have a set of states and in the definition of infinite lists states were never mentioned. But given a list σ what can we observe? We can observe whether σ is empty or not. If it is not empty, we can observe the head element and we are given the rest of the list. So, in fact we can think as the list and all its suffixes as possible states. At each state we can observe an element (or emptiness) and move to the next state (or stop). In a nutshell:

	states	observations	transition dynamics
`list`	$X \subseteq$`list`	empty: $X \to 1$ or hd: $X \to A$	tl: $X \to X$
DFA	S	isfinal: $S \to 2$	next: $S \to S^A$

Putting the two structures side by side exposes the similarity: both DFA and lists are of the form $(Q, Q \to \mathcal{T}(Q))$ where $\mathcal{T}$ is the type of the dynamics present in the system (both in terms of observations and transitions; in the case of DFA we omit the initial state). In the case of lists $\mathcal{T}(Q) = 1 + A \times X$ whereas in the case of DFA we have $\mathcal{T}(Q) = 2 \times X^A$, where $+$ denotes disjoint union of sets and $\times$ cartesian product. Objects of the form $(Q, Q \to \mathcal{T}(Q))$ are precisely $\mathcal{T}$-coalgebras. Formally, Q is an object in a category $\mathbb{C}$ and $\mathcal{T}$ is a an endofunctor in the same category. For the purpose of this note we will mostly think of Q as being a set (an element of the category of sets and functions) and $\mathcal{T}$ a polynomial set functor (built from standard set operations, such as cartesian product, disjoint union, etc).

The strength of the coalgebraic framework lies in the fact that the type $\mathcal{T}$ is rich enough to provide a series of canonical notions, including equivalence (coalgebraic bisimulation) and representatives of behaviour (the so-called final coalgebra). For instance, the final coalgebra of $\mathcal{T}(Q) = 2 \times X^A$ is the set of languages $\mathcal{P}(A^*)$ over the alphabet A, which coincides with the usual semantics defined for automata. Coalgebraic bisimulation also coincides with the standard notion of language equivalence.

The observation that canonical notions of behaviour and equivalence can be derived parametric on the type $\mathcal{T}$ formed the start of the coalgebraic method. More recent research has shown that the type $\mathcal{T}$ is also rich enough to derive specification languages, modal logics, axiomatizations, and algorithms [Cîrstea et al. 2011; Silva 2010].

Having a modular perspective on different models has numerous advantages. First, it provides important bridges between models and even research areas. A recent example of this is the development of NetKAT, a specification and programming language for software-defined networks [Foster et al. 2015; Anderson et al. 2014]. The coalgebraic perspective opened the door to connect work on automata theory and network languages, which guided the definition of the language and corresponding semantics. More interestingly, it also yielded a very efficient procedure for deciding equivalence of NetKAT policies. A second advantage is that many notions, techniques, and algorithms can also be developed at a higher level and then instantiated to the concrete example models. One technique of particular importance is coinduction: the proof principle associated with coalgebraic bisimulation. Coinduction is dual to the well-known principle of induction and it provides a powerful reasoning technique to study equivalence of possibly infinite behaviours. Recent years have witnessed a wide range of articles in exploring enhancements of the coinduction proof method [Pous 2008; Rot et al. 2013; Rot et al. 2014; Bonchi and Pous 2015] increasing even further its applicability. A particular beautiful example of the advantages of the coalgebraic perspective is the recent work of Bonchi&Pous [Bonchi and Pous 2015], featured on the January 2015 cover of CACM, studying an enhanced coinduction method to decide language equivalence for non-deterministic finite automata (NFA). The proposed algorithm improved the classical result of Hopcroft&Karp from the 70's. Their work vividly demonstrates that even classical, well-studied problems like NFA equivalence, can still offer surprising research opportunities, and new ideas that may lead to elegant algorithmic improvements of practical importance.

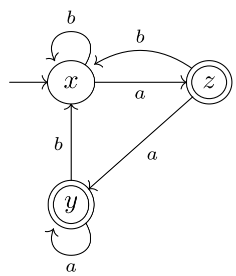
The algorithmic aspects of coalgebra and coinduction have been a quite active recent interest in the community. Understanding existing algorithms and how this abstract perspective can enable development of analogous algorithms for other models has been explored for several algorithms. One such example is Brzozowski's minimization algorithm. In this short note we will explain Brzozowski's algorithm from a coalgebraic perspective and explain how this opens the door to further generalisations without having to re-prove correctness. We will conclude with a discussion on what have been the main lines of research in the coalgebra community and what the future might hold for the field.

2. BRZOZOWSKI'S ALGORITHM, BY EXAMPLE

Brzozowski's minimisation algorithm is one of the simplest minimisation algorithms to explain, though the reason on why it works looks at first sight like magic. Given a (reachable) DFA with state space X the algorithm has four steps.

```
  Brzozowski (X)
(1) reverse and determinize;
(2) take the reachable part;
(3) reverse and determinize;
(4) take the reachable part.
```

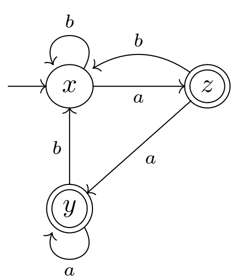
The claim is that the automaton obtained after this procedure is the minimal automaton accepting the same language as the automaton one started with. Let us illustrate the algorithm with an example. Consider the following automaton over the two-letter alphabet $A = \{a, b\}$:



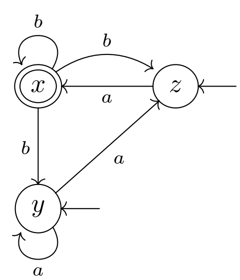
where we denote the final states by double circles and we mark a distinguished initial state with an unlabelled incoming arrow. In the above automaton the state space is $\{x, y, z\}$ where x is the initial state and y and z are final states. It is not too difficult to see that state x accepts all words that end with an a: $L(x) = \{a, b\}^* a$. The above automaton is reachable (all states can be reached from x) but it is not minimal because states y and z accept the same language, namely: $L(y) = \varepsilon + \{a, b\}^* a = L(z)$.

Let us now execute the steps of Brzozowski's algorithm. First, we reverse all transitions and swap initial and final states.

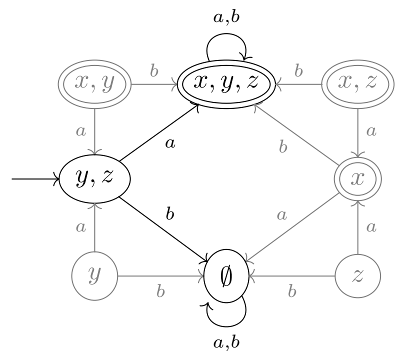
$X =$



$rev(X) =$



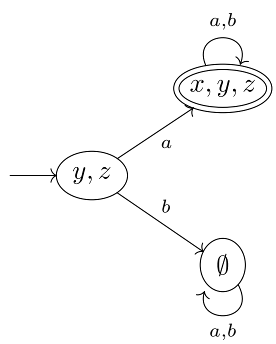
The resulting automaton $rev(X)$ is non-deterministic and hence we construct the corresponding deterministic automaton using the usual subset contruction:



Here, we depict in gray the unreachable states (hence already applying step (2) of the algorithm). The subset construction builds an automaton where the new state space consists of sets of states from the original automaton. The initial state is the set of initial states – $\{y, z\}$ – and a subset V is a final state in the new automaton if $x \in V$ (since x was the only final state in the reversed automaton). Transitions are computed by collecting possible transitions of the non-deterministic automaton:

$$V \xrightarrow{\;a\;} W \qquad W = \{w \mid \; v \xrightarrow{\;a\;} w \;, v \in V\}$$

All in all, after two steps of the algorithm we end up with the automaton

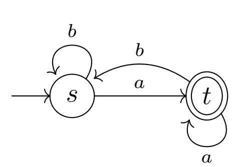


Now note that this new automaton accepts the reverse of the language accepted by X, namely all words starting with an a:

$$L(\,det(rev(X))\,) \;=\; a\,\{a, b\}^* \;=\; reverse(\,L(X)\,)$$

More surprisingly, it is minimal: no two states accept the same language.

Repeating the same two steps on the 3-state automaton above we obtain the following automaton



which is minimal and accepts $\{a, b\}^*\, a$.

Brzozowski's algorithm has attracted some attention in the last couple of years with several papers proposing alternative proofs of correctness and generalisations [Bonchi et al. 2012; Bezhanishvili et al. 2012; Kiefer and Wachter 2014; Bonchi et al. 2014]. This renewed attention was sparked by Prakash Panangaden, SIGLOG's first chair, who during his sabbatical in Oxford gave a nice exposition about a probabilistic extension of the algorithm that he had been working on with his colleagues at McGill [Dinculescu et al. 2011].

Why does the algorithm work? What is the crucial insight that opens the door to further generalisations? We will now briefly describe the ideas behind the proofs presented in [Bonchi et al. 2012; Bezhanishvili et al. 2012].

The main observation in [Bonchi et al. 2012] is that steps `(1)` (and `(3)`) can in fact be captured by using a single well-known categorical construction.

```
 Brzozowski (X)
(1) reverse and determinize;
(2) take the reachable part;
(3) reverse and determinize;
(4) take the reachable part.
```

In particular, let $t\colon X \to X^A$ be the transition function of a deterministic automaton. The transition function $\hat{t}$ of the reversed and determinized automaton can be defined as

$$\begin{array}{l} \hat{t}\colon \mathcal{P}(X) \to \mathcal{P}(X)^A \\ \hat{t}(U)(a) = a-\text{predecessors of } U = \bigcup_{u\in U} \{x \in X \mid t(u)(a) = x\} \end{array}$$

Another way to equivalently write this function is by using the characteristic representation of a set.

$$\begin{array}{l} \hat{t}\colon 2^X \to (2^X)^A \\ \hat{t}(\varphi)(a)(x) = \varphi(t(x)(a)) \end{array}$$

In fact, this definition can be obtained in 3 steps out of which the first and last are just (un)currying (and hence isomorphisms).

$$\begin{tikzcd} X \arrow[d, "t"'] \arrow[r, phantom, "\Vert"] & X\times A \arrow[d] \arrow[r, phantom, "\overset{2^{(-)}}{\longmapsto}"] & 2^{X\times A} \arrow[r, phantom, "\Vert"] & (2^X)^A \\ X^A \arrow[r, phantom, "\Vert"] & X & 2^X \arrow[u] \arrow[r, phantom, "\Vert"] & 2^X \arrow[u, "\hat{t}"] \end{tikzcd}$$

The middle step, which we denote here by $2^{(-)}$ is the application of the *contravariant powerset functor* which is defined as follows, for a set X and a function $f\colon X \to Y$.

$$2^X = \{\varphi \mid \varphi\colon X \to 2\} \cong \{U \mid U \subseteq X\} \qquad \begin{array}{l} 2^f\colon 2^Y \to 2^X \\ 2^f(\varphi) = \varphi \circ f \end{array}$$

The core property of $2^{(-)}$ that justifies the correctness of the algorithm is the fact that $2^{(-)}$ maps surjective functions to injective functions. In the algorithm this corresponds to mapping a reachable automaton to an observable automaton (where no two states accept the same language) which will accept the reverse language.

Describing *reverse and determinize* using the contravariant powerset functor is also the crucial observation for generalizations. The above definition of $2^{(-)}$ can for instance trivially be generalised to any output set B:

$$B^X = \{\phi \mid \phi\colon X \to B\} \qquad \begin{array}{l} B^f\colon B^Y \to B^X \\ B^f(\varphi) = \varphi \circ f \end{array}$$

This means that one can now easily describe Brzozowski's algorithm for Moore automata: $X \longrightarrow B \times X^A$. And the important remark is that the proof of correctness does not need to be changed.

This example gives the indication that the type of the coalgebra under study – for DFA $2 \times (-)^A$ and for Moore automata $B \times (-)^A$ – is rich enough also to derive algorithms. Modular derivation of algorithms where both proofs of correctness and complexity analysis can be done from the type of the coalgebra would be a major advance for the field, as we discuss below.

This generalisation to Moore automata might seem mild but in fact it has shown to be extremely useful in various applications stemming from concurrency theory, program verification and software-defined networks. In particular, Moore automata (for an appropriate output set B) are the automata type needed to capture various equivalences of interest in concurrency theory (most of which is included in the van Glabbeek spectrum plus must/may semantics [Bonchi et al. 2012; Bonchi et al. 2013]). Moore automata are also the models corresponding to KAT [Kozen 1997], a program logic of interest for program verification and compiler optimisation, and NetKAT [Foster et al. 2015; Anderson et al. 2014] an extension of KAT to specify and reason about software-defined networks. Interestingly, the Moore automaton functor can also be defined in the category of vector spaces (B has to be a field) and this then yields the so-called *linear weighted automata* [Schützenberger 1961; Boreale 2009] and a corresponding Brzozowski algorithm. Again, the proof of correctness does not need to be changed because the category of vector spaces has the necessary properties that were used from the category of Sets and functors in the proof presented in [Bonchi et al. 2012] (namely epi-mono factorizations).

A more abstract perspective on the correctness of Brzozowski's algorithm was explored in [Bezhanishvili et al. 2012; Bonchi et al. 2014], where dual adjunctions of automata were used in order to recover the algorithm and also prove its correcteness[1].

There is an adjunction between Set, the category of sets and functions, and $\mathsf{Set}^{\mathrm{op}}$ induced by the contravariant powerset functor $2^{(-)}\colon \mathsf{Set} \to \mathsf{Set}^{\mathrm{op}}$:

$$\begin{tikzcd} \mathsf{Set} \arrow[r, bend left, "2^-"] \arrow[r, phantom, "\perp"] & \mathsf{Set}^{\mathrm{op}} \arrow[l, bend left, "2^-"] \end{tikzcd}$$

This dual adjunction gives a one-to-one correspondence between morphisms $f\colon X \to 2^Y$ in Set and morphisms $\hat{f}\colon 2^X \to Y$ in $\mathsf{Set}^{\mathrm{op}}$ (that is, $\hat{f}\colon Y \to 2^X$ in Set). This correspondence is given by the following equation

$$y \in f(x) \iff x \in \hat{f}(y)$$

[1]To be precise, in [Bezhanishvili et al. 2012] the authors only explored algebra-coalgebra duality, but we will here describe the related perspective of [Bonchi et al. 2014].

Brzozowski's algorithm is obtained once one observes that the above dual adjunction actual lifts to a suitable category of automata

$$\begin{tikzcd} \mathsf{Aut}(L) \arrow[r, bend left, "\mathbf{2}^{-}"] \arrow[r, phantom, "\perp"] \arrow[d] & \mathsf{Aut}(rev(L))^{\mathrm{op}} \arrow[l, bend left, "\mathbf{2}^{-}"] \arrow[d] \\ \mathsf{Set} \arrow[r, bend left, "2^{-}"] \arrow[r, phantom, "\perp"] & \mathsf{Set}^{\mathrm{op}} \arrow[l, bend left, "2^{-}"] \end{tikzcd}$$

Starting with an automaton $\mathcal{A}$ accepting a language L (an object in $\mathsf{Aut}(L)$) we obtain an automaton $\mathcal{A}^r = \mathbf{2}(\mathcal{A})$ accepting the reverse language $rev(L)$ (an object in $\mathsf{Aut}(rev(L))$). If $\mathcal{A}$ is a reachable automaton then $\mathbf{2}(\mathcal{A})$ will be an observable automaton (that is, no two states accept the same language). If we now take reachability and apply $\mathbf{2}^{-}\colon \mathsf{Set}^{\mathrm{op}} \to \mathsf{Set}$ to obtain an observable automaton accepting $rev(rev(L)) = L$.

Reachability and observability can be described in this setting as universal maps. In a category initial and final objects play an important role. An object $\mathcal{I}$ is initial if there exists a unique map $\mathcal{I} \to O$ into any other object O in the category. Dually, an object $\mathcal{Z}$ is final if there exists a unique map $O \to \mathcal{Z}$ from any other object O into $\mathcal{Z}$. Initial and final objects are intimately connected to definitions of (co)algebraic datatypes in functional languages and also to the notions of induction and coinduction, respectively. We will not explore this here, but instead will use the universal properties of initial and final objects in the category of automata to define reachability and observability.

Let $\mathcal{I}$ and $\mathcal{Z}$ be, respectively, the initial and final objects in $\mathsf{Aut}(L)$ (or $\mathsf{Aut}(rev(L))$) and let $\mathcal{X}$ be an automaton. $\mathcal{X}$ is reachable if the unique morphism from the initial object $r\colon \mathcal{I} \to \mathcal{X}$ is surjective. $\mathcal{X}$ is observable if the final morphism $o\colon \mathcal{X} \to \mathcal{Z}$ is injective. The initial object morphism $r\colon \mathcal{I} \to \mathcal{X}$ is mapped by $\mathbf{2}^{-}$ to the final morphism $o\colon \mathbf{2}^{\mathcal{X}} \to \mathcal{Z}$ in $\mathsf{Aut}(rev(L))$. Consequently, if $\mathcal{X}$ is reachable then $\mathcal{A}^r = \mathbf{2}(\mathcal{X})$ is observable.

3. DISCUSSION

Coalgebra has been a very active research area in the last few decades. The dedicated workshop – *Coalgebraic Methods in Computer Science* – was organised for the first time when ETAPS started, in 1998, by Bart Jacobs, Larry Moss, Horst Reichel, and Jan Rutten. Since then, it has always been collocated with ETAPS, becoming bi-annual since the start of CALCO (Conference on Algebra and Coalgebra) in 2005 (CALCO has been organised as independent event, with the exception of the upcoming 2015 edition in which it will be co-located with MFPS XXXI, the 31st edition of the conference on Mathematical Foundations of Programming Semantics). In 2010, there was a special celebration for the 10th edition of CMCS, with four invited talks covering the main areas of research:

— Probabilistic systems coalgebraically (Ana Sokolova)
— Logic and coalgebra (Dirk Pattinson)
— Operational semantics coalgebraically (Bartek Klin)
— Coalgebra in functional programming and type theory (Venanzio Capretta)

After the workshop, all invited speakers wrote overview articles which were published in a special issue [Jacobs et al. 2011]. These four areas have been key in the widespread of coalgebra in other areas like concurrency theory (e.g. [Mislove et al. 2004; Hasuo 2010; Schröder and Venema 2010; Kerstan and König 2012; Madiot et al. 2014; Urabe and Hasuo 2014]), programming language semantics (e.g. [Bonchi and Pous 2013; Abel et al. 2013; Staton and Levy 2013; Jeannin et al. 2013; Bonchi et al. 2015; Foster et al. 2015; Staton 2015; Anderson et al. 2014]), and artificial intelligence (e.g. [Goré et al. 2010; Schröder and Pattinson 2011; Kulacka et al. 2013]).

In the last five years, there has been a large body of research in applying (co)algebraic methods to automata theory (e.g. [Bonsangue et al. 2013; Winter et al. 2013; Rutten et al. 2013; Silva et al. 2013; Adámek et al. 2014a; Adámek et al. 2014b; Goncharov et al. 2014]). One thing that became evident in this line of research was that the combination of coalgebraic and algebraic methods is essential in order to fully understand automata constructions and algorithms. In my personal opinion, the synergy between algebra and coalgebra has a lot of potential to generate new results and algorithmic insights. I think that a framework based solely on just either coalgebra or algebra will not explore the full potential of abstraction offered by both frameworks.

Apart from the applications in automata theory I think there are two directions that show promising challenges for the theory of coalgebras and in which I believe coalgebra could have significant impact.

The first direction concerns the applications of coinduction in the context of verification and, in particular, quantitative verification. To achieve the verification of real-world systems, the study of efficient coinductive techniques for specifying and reasoning about dynamical systems is needed. A system can be expressed as a coalgebra, and its desired observable behaviour can be expressed as a (generalised) regular expression. The correctness of the system can then be verified by proving that these two formalisms are observably equivalent or, more realistically, that one is a refinement of the other. Coalgebra has mainly focused on techniques for equivalence, and lifting the general theory to handle simulation/refinement is a pressing research task that has received recent interest [Hasuo 2010; Urabe and Hasuo 2014] but is far from finished. One needs efficient algorithms for the calculation of a witness of the equivalence or refinement, which in the coalgebraic approach consists of a suitable (bi)simulation relation. Already for ordinary regular expressions, a naive implementation can be exponentially complex, even if most of the operations are polynomial. However, very efficient algorithms exist to compositionally derive a finite automaton from ordinary regular expressions. Extending these algorithms to the generalised algebraic and coalgebraic framework of regular expressions and implement them in existing tools would be an important step in enabling the application to industrial case studies. In recent work on NetKAT [Anderson et al. 2014], an instance of a generalised bisimulation procedure provides an efficient algorithm for a specification language for software-defined networking (SDN), which brings coalgebra closer to a more realistic application domain in specification and verification.

In the context of this first line of research, having the right notion of equivalence is crucial. Defining the behaviour or the behavioural similarity of two systems that behave the same has been a fundamental issue in computer science. Many different equivalences, usually referred as behavioural equivalences, have been introduced for different kinds of systems (e.g., functional, non-deterministic and concurrent systems). A huge corpus of conceptual and computational tools consisting of algorithms, proof techniques, rule formats, and languages have been developed to reason about systems and their behavioural equivalences.

The growing interest in quantitative properties has motivated the definition of quantitative models and related notions of behavioural equivalences. For many quantitative properties, standard notions of equivalence are inadequate since they only allow Boolean reasoning: either two systems are equivalent or they are not. As an example, consider a system consisting of a producer, a consumer, and a buffer connecting them: whenever the buffer is full, the newly-produced resources get lost. Suppose that the producer or the consumer are complex networks, and suppose that we know by empirical data their rates of production and consumption (examples of this kind are widely studied in the performance modelling of systems). A probabilistic model of this system allows for several interesting quantitative questions, like: How far is the system with a buffer of capacity n from the ideal system that never loses resources? For which values of n is the distance between the behaviours of these two systems smaller than a threshold ε? Motivated by these considerations, many authors have recently proposed several behavioural metrics or closely related notions [Desharnais et al. 1999; van Breugel and Worrell 2006; Cerný et al. 2010; Bacci et al. 2013]. Unlike ordinary equivalences, behavioural metrics express the distance between the behaviours of systems: if the distance is zero, then the systems are behaviourally equivalent. These works show the effectiveness of behavioural metrics as foundations of quantitative reasoning, but there is much work still to be done. In particular, many conceptual and computational tools are not yet available for metrics. There is a pressing need of lifting such tools from equivalences to metrics, which I think can be achieved in a modular way using the theory of coalgebras.

The second direction is in the area of automata learning. Learning techniques have become increasingly important for their applications to a wide variety of software engineering problems, especially in the analysis and testing of complex systems. Recently, they have been successfully applied to security protocol testing, the analysis of botnet command and control protocols, in regression testing of telecommunication protocols, and conformance testing of communication protocols. For each application domain, the type of automaton to be learned varies, hence the algorithm needs to be adapted and this generates challenging problems. For all the proposed algorithms, which followed the seminal work of Angluin in 1987 [Angluin 1987], all the correctness and complexity results need to be re-proved for every different case, even if the model under study is not a major variation on deterministic automata. Similar observations apply to the area of Bayesian or probabilistic learning, where the development of algorithms is done on a call-by-need basis. I believe that coalgebra provides the perfect framework for the systematic study and development of learning algorithms in automata and Bayesian networks. Recent preliminary work [Jacobs and Silva 2014] shows the feasibility and the potential of the approach. There are a lot of challenges to be tackled, which is also a source of inspiration for further research in the coalgebra community, but the point in the horizon is to provide a strong mathematical foundation and bring compositional techniques to learning algorithms.

REFERENCES

Andreas Abel, Brigitte Pientka, David Thibodeau, and Anton Setzer. 2013. Copatterns: Programming Infinite Structures by Observations. In *Proceedings of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '13)*. ACM, New York, NY, USA, 27–38. DOI:http://dx.doi.org/10.1145/2429069.2429075

Jirí Adámek, Stefan Milius, Robert S. R. Myers, and Henning Urbat. 2014a. Generalized Eilenberg Theorem I: Local Varieties of Languages. In *Foundations of Software Science and Computation Structures - 17th International Conference, FOSSACS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings (Lecture Notes in Computer Science)*, Anca Muscholl (Ed.), Vol. 8412. Springer, 366–380. DOI:http://dx.doi.org/10.1007/978-3-642-54830-7_24

Jirí Adámek, Robert S. R. Myers, Henning Urbat, and Stefan Milius. 2014b. On Continuous Nondeterminism and State Minimality. *Electr. Notes Theor. Comput. Sci.* 308 (2014), 3–23. DOI:http://dx.doi.org/10.1016/j.entcs.2014.10.002

Carolyn Jane Anderson, Nate Foster, Arjun Guha, Jean-Baptiste Jeannin, Dexter Kozen, Cole Schlesinger, and David Walker. 2014. NetkAT: semantic foundations for networks. In *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego,* *CA, USA, January 20-21, 2014*, Suresh Jagannathan and Peter Sewell (Eds.). ACM, 113–126. DOI:http://dx.doi.org/10.1145/2535838.2535862

Dana Angluin. 1987. Learning Regular Sets from Queries and Counterexamples. *Inf. Comput.* 75, 2 (1987), 87–106. DOI:http://dx.doi.org/10.1016/0890-5401(87)90052-6

Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, and Radu Mardare. 2013. On-the-Fly Exact Computation of Bisimilarity Distances. In *Tools and Algorithms for the Construction and Analysis of Systems - 19th International Conference, TACAS 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings (Lecture Notes in Computer Science)*, Nir Piterman and Scott A. Smolka (Eds.), Vol. 7795. Springer, 1–15. DOI:http://dx.doi.org/10.1007/978-3-642-36742-7_1

Paolo Baldan and Daniele Gorla (Eds.). 2014. *CONCUR 2014 - Concurrency Theory - 25th International Conference, CONCUR 2014, Rome, Italy, September 2-5, 2014. Proceedings*. Lecture Notes in Computer Science, Vol. 8704. Springer. DOI:http://dx.doi.org/10.1007/978-3-662-44584-6

Jon Barwise and Larry Moss. 1996. *Vicious Circles*. Center for the Study of Language and Information - Stanford.

Nick Bezhanishvili, Clemens Kupke, and Prakash Panangaden. 2012. Minimization via Duality. In *Logic, Language, Information and Computation - 19th International Workshop, WoLLIC 2012, Buenos Aires, Argentina, September 3-6, 2012. Proceedings (Lecture Notes in Computer Science)*, C.-H. Luke Ong and Ruy J. G. B. de Queiroz (Eds.), Vol. 7456. Springer, 191–205. DOI:http://dx.doi.org/10.1007/978-3-642-32621-9_14

Filippo Bonchi, Marcello M. Bonsangue, Georgiana Caltais, Jan J. M. M. Rutten, and Alexandra Silva. 2012. Final Semantics for Decorated Traces. *Electr. Notes Theor. Comput. Sci.* 286 (2012), 73–86. DOI:http://dx.doi.org/10.1016/j.entcs.2012.08.006

Filippo Bonchi, Marcello M. Bonsangue, Helle Hvid Hansen, Prakash Panangaden, Jan J. M. M. Rutten, and Alexandra Silva. 2014. Algebra-coalgebra duality in Brzozowski's minimization algorithm. *ACM Trans. Comput. Log.* 15, 1 (2014), 3. DOI:http://dx.doi.org/10.1145/2490818

Filippo Bonchi, Marcello M. Bonsangue, Jan J. M. M. Rutten, and Alexandra Silva. 2012. Brzozowski's Algorithm (Co)Algebraically. In *Logic and Program Semantics - Essays Dedicated to Dexter Kozen on the Occasion of His 60th Birthday (Lecture Notes in Computer Science)*, Robert L. Constable and Alexandra Silva (Eds.), Vol. 7230. Springer, 12–23. DOI:http://dx.doi.org/10.1007/978-3-642-29485-3_2

Filippo Bonchi, Georgiana Caltais, Damien Pous, and Alexandra Silva. 2013. Brzozowski's and Up-To Algorithms for Must Testing. In *Programming Languages and Systems - 11th Asian Symposium, APLAS 2013, Melbourne, VIC, Australia, December 9-11, 2013. Proceedings (Lecture Notes in Computer Science)*, Chung-chieh Shan (Ed.), Vol. 8301. Springer, 1–16. DOI:http://dx.doi.org/10.1007/978-3-319-03542-0_1

Filippo Bonchi and Damien Pous. 2013. Checking NFA equivalence with bisimulations up to congruence, See Giacobazzi and Cousot [2013], 457–468. DOI:http://dx.doi.org/10.1145/2429069.2429124

Filippo Bonchi and Damien Pous. 2015. Hacking nondeterminism with induction and coinduction. *Commun. ACM* 58, 2 (2015), 87–95. DOI:http://dx.doi.org/10.1145/2713167

Filippo Bonchi, Pawel Sobocinski, and Fabio Zanasi. 2015. Full Abstraction for Signal Flow Graphs, See Rajamani and Walker [2015], 515–526. DOI:http://dx.doi.org/10.1145/2676726.2676993

Marcello M. Bonsangue, Stefan Milius, and Alexandra Silva. 2013. Sound and Complete Axiomatizations of Coalgebraic Language Equivalence. *ACM Trans. Comput. Log.* 14, 1 (2013), 7. DOI:http://dx.doi.org/10.1145/2422085.2422092

Michele Boreale. 2009. Weighted Bisimulation in Linear Algebraic Form. In *CONCUR 2009 - Concurrency Theory, 20th International Conference, CONCUR 2009, Bologna, Italy, September 1-4, 2009. Proceedings (Lecture Notes in Computer Science)*, Mario Bravetti and Gianluigi Zavattaro (Eds.), Vol. 5710. Springer, 163–177. DOI:http://dx.doi.org/10.1007/978-3-642-04081-8_12

Pavol Cerný, Thomas A. Henzinger, and Arjun Radhakrishna. 2010. Simulation Distances, See Gastin and Laroussinie [2010], 253–268. DOI:http://dx.doi.org/10.1007/978-3-642-15375-4_18

Corina Cîrstea, Alexander Kurz, Dirk Pattinson, Lutz Schröder, and Yde Venema. 2011. Modal Logics are Coalgebraic. *Comput. J.* 54, 1 (2011), 31–41. DOI:http://dx.doi.org/10.1093/comjnl/bxp004

Josee Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. 1999. Metrics for Labeled Markov Systems. In *CONCUR '99: Concurrency Theory, 10th International Conference, Eindhoven, The Netherlands, August 24-27, 1999, Proceedings (Lecture Notes in Computer Science)*, Jos C. M. Baeten and Sjouke Mauw (Eds.), Vol. 1664. Springer, 258–273. DOI:http://dx.doi.org/10.1007/3-540-48320-9_19

Monica Dinculescu, Christopher Hundt, Prakash Panangaden, Joelle Pineau, and Doina Precup. 2011. The Duality of State and Observation in Probabilistic Transition Systems. In *Logic, Language, and Computation - 9th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2011, Kutaisi, Georgia, September 26-30, 2011, Revised Selected Papers (Lecture Notes in Computer Science)*, Guram Bezhanishvili, Sebastian Löbner, Vincenzo Marra, and Frank Richter (Eds.), Vol. 7758. Springer, 206–230. DOI:http://dx.doi.org/10.1007/978-3-642-36976-6_14

Nate Foster, Dexter Kozen, Matthew Milano, Alexandra Silva, and Laure Thompson. 2015. A Coalgebraic Decision Procedure for NetKAT, See Rajamani and Walker [2015], 343–355. DOI:http://dx.doi.org/10.1145/2676726.2677011

Paul Gastin and François Laroussinie (Eds.). 2010. *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*. Lecture Notes in Computer Science, Vol. 6269. Springer. DOI:http://dx.doi.org/10.1007/978-3-642-15375-4

Roberto Giacobazzi and Radhia Cousot (Eds.). 2013. *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13, Rome, Italy - January 23 - 25, 2013*. ACM. http://dl.acm.org/citation.cfm?id=2429069

Sergey Goncharov, Stefan Milius, and Alexandra Silva. 2014. Towards a Coalgebraic Chomsky Hierarchy - (Extended Abstract). In *Theoretical Computer Science - 8th IFIP TC 1/WG 2.2 International Conference, TCS 2014, Rome, Italy, September 1-3, 2014. Proceedings (Lecture Notes in Computer Science)*, Josep Diaz, Ivan Lanese, and Davide Sangiorgi (Eds.), Vol. 8705. Springer, 265–280. DOI:http://dx.doi.org/10.1007/978-3-662-44602-7_21

Rajeev Goré, Clemens Kupke, Dirk Pattinson, and Lutz Schröder. 2010. Global Caching for Coalgebraic Description Logics. In *Automated Reasoning, 5th International Joint Conference, IJCAR 2010, Edinburgh, UK, July 16-19, 2010. Proceedings (Lecture Notes in Computer Science)*, Jürgen Giesl and Reiner Hähnle (Eds.), Vol. 6173. Springer, 46–60. DOI:http://dx.doi.org/10.1007/978-3-642-14203-1_5

Ichiro Hasuo. 2010. Generic Forward and Backward Simulations II: Probabilistic Simulation, See Gastin and Laroussinie [2010], 447–461. DOI:http://dx.doi.org/10.1007/978-3-642-15375-4_31

Bart Jacobs, Milad Niqui, Jan J. M. M. Rutten, and Alexandra Silva. 2011. Preface. *Theor. Comput. Sci.* 412, 38 (2011), 4967–4968. DOI:http://dx.doi.org/10.1016/j.tcs.2011.06.001

Bart Jacobs, JJMM Rutten, and others. 1997. An introduction to (co) algebra and (co) induction. *EATCS Bulletin* 62 (1997), 222–259.

Bart Jacobs and Alexandra Silva. 2014. Automata Learning: A Categorical Perspective. In *Horizons of the Mind. A Tribute to Prakash Panangaden - Essays Dedicated to Prakash Panangaden on the Occasion of His 60th Birthday (Lecture Notes in Computer Science)*, Franck van Breugel, Elham Kashefi, Catuscia Palamidessi, and Jan Rutten (Eds.), Vol. 8464. Springer, 384–406. DOI:http://dx.doi.org/10.1007/978-3-319-06880-0_20

Jean-Baptiste Jeannin, Dexter Kozen, and Alexandra Silva. 2013. Language Constructs for Non-Well-Founded Computation. In *Programming Languages and Systems - 22nd European Symposium on Programming, ESOP 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings (Lecture Notes in Computer Science)*, Matthias Felleisen and Philippa Gardner (Eds.), Vol. 7792. Springer, 61–80. DOI:http://dx.doi.org/10.1007/978-3-642-37036-6_4

Henning Kerstan and Barbara König. 2012. Coalgebraic Trace Semantics for Probabilistic Transition Systems Based on Measure Theory. In *CONCUR 2012 - Concurrency Theory - 23rd International Conference, CONCUR 2012, Newcastle upon Tyne, UK, September 4-7, 2012. Proceedings (Lecture Notes in Computer Science)*, Maciej Koutny and Irek Ulidowski (Eds.), Vol. 7454. Springer, 410–424. DOI:http://dx.doi.org/10.1007/978-3-642-32940-1_29

Stefan Kiefer and Björn Wachter. 2014. Stability and Complexity of Minimising Probabilistic Automata. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II (Lecture Notes in Computer Science)*, Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias (Eds.), Vol. 8573. Springer, 268–279. DOI:http://dx.doi.org/10.1007/978-3-662-43951-7_23

Dexter Kozen. 1997. Kleene Algebra with Tests. *ACM Trans. Program. Lang. Syst.* 19, 3 (1997), 427–443. DOI:http://dx.doi.org/10.1145/256167.256195

Agnieszka Kulacka, Dirk Pattinson, and Lutz Schröder. 2013. Syntactic Labelled Tableaux for Lukasiewicz Fuzzy ALC. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, Francesca Rossi (Ed.). IJCAI/AAAI. http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6831

Jean-Marie Madiot, Damien Pous, and Davide Sangiorgi. 2014. Bisimulations Up-to: Beyond First-Order Transition Systems, See Baldan and Gorla [2014], 93–108. DOI:http://dx.doi.org/10.1007/978-3-662-44584-6_8

Michael W. Mislove, Joël Ouaknine, Dusko Pavlovic, and James Worrell. 2004. Duality for Labelled Markov Processes. In *Foundations of Software Science and Computation Structures, 7th International Conference, FOSSACS 2004, Held as Part of the Joint European Conferences on Theory* *and Practice of Software, ETAPS 2004, Barcelona, Spain, March 29 - April 2, 2004, Proceedings (Lecture Notes in Computer Science)*, Igor Walukiewicz (Ed.), Vol. 2987. Springer, 393–407. DOI:http://dx.doi.org/10.1007/978-3-540-24727-2_28

Damien Pous. 2008. *Techniques modulo pour les bisimulations*. Ph.D. Dissertation. PhD thesis, ENS Lyon.

Sriram K. Rajamani and David Walker (Eds.). 2015. *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015*. ACM. http://dl.acm.org/citation.cfm?id=2676726

Jurriaan Rot, Filippo Bonchi, Marcello Bonsangue, Damien Pous, JJMM Rutten, and Alexandra Silva. 2014. Enhanced coalgebraic bisimulation. *Mathematical Structures in Computer Science (to appear, 2014)* (2014).

Jurriaan Rot, Marcello M. Bonsangue, and Jan J. M. M. Rutten. 2013. Coalgebraic Bisimulation-Up-To. In *SOFSEM 2013: Theory and Practice of Computer Science, 39th International Conference on Current Trends in Theory and Practice of Computer Science, Špindlerův Mlýn, Czech Republic, January 26-31, 2013. Proceedings (Lecture Notes in Computer Science)*, Peter van Emde Boas, Frans C. A. Groen, Giuseppe F. Italiano, Jerzy R. Nawrocki, and Harald Sack (Eds.), Vol. 7741. Springer, 369–381. DOI:http://dx.doi.org/10.1007/978-3-642-35843-2_32

Jan Rutten, Adolfo Ballester-Bolinches, and Enric Cosme-Llópez. 2013. Varieties and Covarieties of Languages (Extended Abstract). *Electr. Notes Theor. Comput. Sci.* 298 (2013), 7–28. DOI:http://dx.doi.org/10.1016/j.entcs.2013.09.005

Jan J. M. M. Rutten. 2000. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.* 249, 1 (2000), 3–80. DOI:http://dx.doi.org/10.1016/S0304-3975(00)00056-6

Lutz Schröder and Dirk Pattinson. 2011. Description Logics and Fuzzy Probability. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, Toby Walsh (Ed.). IJCAI/AAAI, 1075–1081. http://ijcai.org/papers11/Papers/IJCAI11-184.pdf

Lutz Schröder and Yde Venema. 2010. Flat Coalgebraic Fixed Point Logics, See Gastin and Laroussinie [2010], 524–538. DOI:http://dx.doi.org/10.1007/978-3-642-15375-4_36

Marcel Paul Schützenberger. 1961. On the Definition of a Family of Automata. *Information and Control* 4, 2-3 (1961), 245–270. DOI:http://dx.doi.org/10.1016/S0019-9958(61)80020-X

Alexandra Silva. 2010. *Kleene Coalgebra*. Ph.D. Dissertation. Radboud University Nijmegen.

Alexandra Silva, Filippo Bonchi, Marcello M. Bonsangue, and Jan J. M. M. Rutten. 2013. Generalizing determinization from automata to coalgebras. *Logical Methods in Computer Science* 9, 1 (2013). DOI:http://dx.doi.org/10.2168/LMCS-9(1:9)2013

Sam Staton. 2015. Algebraic Effects, Linearity, and Quantum Programming Languages, See Rajamani and Walker [2015], 395–406. DOI:http://dx.doi.org/10.1145/2676726.2676999

Sam Staton and Paul Blain Levy. 2013. Universal properties of impure programming languages, See Giacobazzi and Cousot [2013], 179–192. DOI:http://dx.doi.org/10.1145/2429069.2429091

Natsuki Urabe and Ichiro Hasuo. 2014. Generic Forward and Backward Simulations III: Quantitative Simulations by Matrices, See Baldan and Gorla [2014], 451–466. DOI:http://dx.doi.org/10.1007/978-3-662-44584-6_31

Franck van Breugel and James Worrell. 2006. Approximating and computing behavioural distances in probabilistic transition systems. *Theor. Comput. Sci.* 360, 1-3 (2006), 373–385. DOI:http://dx.doi.org/10.1016/j.tcs.2006.05.021

Joost Winter, Marcello M. Bonsangue, and Jan J. M. M. Rutten. 2013. Coalgebraic Characterizations of Context-Free Languages. *Logical Methods in Computer Science* 9, 3 (2013). DOI:http://dx.doi.org/10.2168/LMCS-9(3:14)2013

VERIFICATION COLUMN

NEHA RUNGTA, SGT Inc./NASA Ames Research Center
neha.s.rungta@nasa.gov

In this issue of the SIGLOG newsletter, the technical column on verification presents an invited contribution on "Algorithmic Logic-Based Verification" by Arie Gurfinkel (Software Engineering Institute / Carnegie Mellon University), Temesghen Kahsai (NASA Ames / Carnegie Mellon University), and Jorge Navas (NASA Ames / SGT Inc). The increasing number of programming languages and intermediate representations poses an obstacle in the development of a unified verification framework. The article presents a case for the use of Constrained Horn Clauses (CHC) as a universal representation for transition systems to overcome this obstacle. This article should be of special interest to the readers of this newsletter as CHCs have been used in various logic-based application domains, but are relatively new to the domain of verification. I would like to thank the authors for submitting the article.

Algorithmic Logic-Based Verification

Arie Gurfinkel
SEI/CMU

Temesghen Kahsai
NASA Ames/CMU

Jorge A. Navas
NASA Ames/SGT

1. INTRODUCTION

Turing in his seminal paper "Checking a Large Routine" [Turing 1949] already asked the question whether it was possible to check a routine was *right*. Among other contributions, he proposed flowcharts as a concise program representation. He also described a method based on the insight that a programmer should make a number of definite assertions which can be proven individually, and from which the correctness of the whole program could easily follow. It took several years until Floyd [Floyd 1967] and Hoare [Hoare 1969], inspired by McCarthy [McCarthy 1963] and Naur [Naur 1966]'s works, established a logic based on a deductive system what is called today *Floyd-Hoare logic* that allowed proving correctness of programs in a rigorous manner. Dijkstra [Dijkstra 1975] presented the first semi-algorithmic view of the Floyd-Hoare logic based on the ideas of *predicate transformers*. Since then, the field of software verification has been growing rapidly during the last decades with many available techniques. Among them, *Abstract Interpretation* [Cousot and Cousot 1977], *Model Checking* [Clarke and Emerson 1981; Queille and Sifakis 1982], and *Symbolic Execution* [King 1976] are probably the most predominant algorithmic (i.e., fully automated) techniques today.

Regardless of the underlying techniques, most software verifiers aim at proving some correctness claims by computing the meaning of the program by either (a) inspecting directly the source code of the program or (b) analyzing some specification describing all program behaviors. Since the problem of computing the meaning of a program is undecidable, most software verifiers offer different trade-offs between completeness, efficiency and accuracy. Therefore, it is highly desirable to combine different techniques to get their maximal advantages. Unfortunately, due to the existence of a myriad of program representations and language specifications, the communication between verifiers is not so simple and the results are often hard to combine and reuse.

In this article, we make a case for *Constrained Horn Clauses (CHCs)*, a fragment of First Order Logic, as the basis for software verification. CHCs are a uniform way to formally represent transition systems while allowing many encoding styles of verification conditions (VCs). Moreover, CHCs allow separating the concerns of the programming

This material is based upon work funded and supported by NASA Contract No. NNX14AI09G, NSF Award No. 1422705 and the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense, NASA or NSF. This material has been approved for public release and unlimited distribution. DM-0002273.

language syntax and the verification techniques. The main idea is that Verification Condition Generators (VCGs) translate the input program together with its assertions into a set of VCs represented by means of CHCs while pure logic-based algorithms using, for instance, abstract interpretation and model checking techniques can focus on solving the CHCs. Finally, CHCs provide a formal logical foundations that simplify the sharing of intermediate results.

Although the use of CHC as the basis to represent transition systems is relatively new in the verification community, CHCs have been used for decades in other fields. For example, CHCs are the basis of *Constraint Logic Programming* (CLP) [Jaffar and Lassez 1987]. CLP has been successfully used in many different contexts such as management decision problems, trading, scheduling, electrical circuit analysis, mapping in genetics, *etc.* (see [Marriott and Stuckey 1998] for a survey). Although the standard execution model for CLP is based on *depth-first search* which is incomplete in presence of recursive CHCs, CLP systems are usually augmented with *tabling* capabilities to record calls and their answers for reuse in future calls that can avoid unnecessary infinite computations.

As a result of the success of CLP and LP (i.e., unconstrained Horn clauses) as programming languages, abstract-interpretation-based static analysis of these languages has been a very active area since the 80's. The primary target is code optimization in LP/CLP compilers (see e.g., [Søndergaard 1986; Bruynooghe et al. 1987; Warren et al. 1988; Muthukumar and Hermenegildo 1989]). CLP has been also used as the basis for software model checking [Delzanno and Podelski 1999; Flanagan 2003; Jaffar et al. 2004] of concurrent and timed automata systems as well as in the context of static analysis of imperative and object-oriented languages (e.g., [Peralta et al. 1998; Méndez-Lojo et al. 2007]).

Therefore, it is a fair question to ask why now this renewed interest in the use of CHCs as the basis of analysis and verification? The answer lies in the new powerful decision engines, called SMT solvers, that have been recently developed and perfected in the verification community. Recently, new SMT-based techniques have emerged (e.g., [Hoder and Bjørner 2012a; Grebenshchikov et al. 2012; Komuravelli et al. 2014]) that are able to automatically solve recursive CHCs which were beyond the capabilities of tabled CLP systems. Moreover, together with smart VC encodings larger systems of CHCs can be now solved much faster. These advances have facilitated the implementation of efficient CHC solvers that can combine many existing verification techniques based on abstract interpretation and model checking in more sophisticated ways and compete with existing state-of-the-art approaches.

To provide a concrete example of a state-of-the-art CHC-based verifier, we present in this article SEAHORN an efficient verification framework. SEAHORN aims at providing developers and researchers a collection of modular and reusable verification components that can reduce the burden of building a new software verifier. Similar to modern compilers, SEAHORN is split into three main components: the front-end, the middle-end, and the back-end.

The front-end deals with the syntax and semantics of the input programming language and generates an internal intermediate representation (IR) more suitable for verification. SEAHORN relies on LLVM's front-ends for this and it uses the LLVM [Lattner and Adve 2004] infrastructure to optimize IR (LLVM bitcode). Although the role of the front-ends are often played down and most research papers tend to omit them, we argue that its role is a predominant one and our experience with SEAHORN demonstrates clearly that the front-end must be a major component in the design of any verifier. Note that with this front-end, SEAHORN does not verify source code but instead the optimized internal representation used by a real compiler (e.g., Clang [1]). Although this is not yet machine code it is a more realistic approach than the one adopted by source code-based verifiers since it takes into consideration the *WYSINWYX (What-You-See-Is-Not-What-You-Execute)* phenomenon. The middle-end uses CHCs to encode the verification conditions that arise from the verification of the LLVM bitcode and it is fully parametric on the semantics used to encode the VCs. SEAHORN provides several out-of-the-box encodings which have been shown useful in practice. Finally, the back-end discharges the verification conditions. Since this is a hard problem SEAHORN uses a variety of state-of-the-art SMT-based model checking and abstract interpretation-based solvers.

This versatile and flexible design not only allows easily interchanging multiple VC encodings and solvers but also it makes possible the verification of new programming languages or language specifications assuming a translation to CHCs is provided. This makes SEAHORN an interesting verification infrastructure that allows developers and researchers experimenting with new techniques.

In spite of the fact that efficiency is not the primary aspect in the design of SEAHORN, it has demonstrated its practicality by its performance at the annual Competition on Software Verification (SV-COMP 2015) [Beyer 2015] as well as a successful experience at verifying industrial software.

2. BACKGROUND

In this section, we describe how verification conditions that arise from a verification problem can be encoded into CHCs so that specialized solvers can check their (un)-satisfiability. This approach has been adopted by an increasing number of verifiers such as Threader [Gupta et al. 2011], UFO [Albarghouthi et al. 2012], SEAHORN [Gurfinkel et al. 2015], HSF [Grebenshchikov et al. 2012], VeriMAP [De Angelis et al. 2014], Eldarica [Rümmer et al. 2013], and TRACER [Jaffar et al. 2012].

2.1. Constrained Horn Clauses

Given the sets $\mathcal{F}$ of function symbols, $\mathcal{P}$ of predicate symbols, and $\mathcal{V}$ of variables, a *Constrained Horn Clause (CHC)* is a formula:

$$\forall \mathcal{V} \cdot (\phi \wedge p_1[X_1] \wedge \cdots \wedge p_k[X_k] \rightarrow h[X]),\ \text{for } k \geq 0$$

where ϕ is a constraint over $\mathcal{F}$ and $\mathcal{V}$ with respect to some background theory; $X_i, X \subseteq \mathcal{V}$ are (possibly empty) vectors of variables; $p_i[X_i]$ is an application $p(t_1, \ldots, t_n)$ of an n-ary predicate symbol $p \in \mathcal{P}$ for first-order terms t_i constructed from $\mathcal{F}$ and X_i; and $h[X]$ is either defined analogously to p_i or is $\mathcal{P}$-free (i.e., no $\mathcal{P}$ symbols occur in h).

Here, h is called the *head* of the clause and $\phi \wedge p_1[X_1] \wedge \cdots \wedge p_k[X_k]$ is called the *body*. A clause is called a *query* if its head is $\mathcal{P}$-free, and otherwise, it is called a *rule*. A rule with body true is called a *fact*. We say a clause is *linear* if its body contains at most one predicate symbol, otherwise, it is called *non-linear*. In this article, we follow the CLP convention of writing Horn clauses as $h[X] \leftarrow \phi, p_1[X_1], \ldots, p_k[X_k]$.

A set of CHCs is satisfiable if there exists an interpretation $\mathcal{J}$ of the predicate symbols $\mathcal{P}$ such that each constraint ϕ is true under $\mathcal{J}$.

2.2. Weakest preconditions calculus

Dijkstra's weakest preconditions calculus [Dijkstra 1975] is a classical method for proving correctness of programs. The main idea is to reduce the problem of verifying a *Hoare triple* $\{Pre\}P\{Post\}$ to proving a pure first-order logic formula by applying a

[1] A C language family front-end for LLVM (http://clang.llvm.org).

$$
\begin{array}{lcl}
\mathtt{wp}(\mathtt{if}\ C\ S_1\ \mathtt{else}\ S_2, \phi) & \rightsquigarrow & (C \wedge \mathtt{wp}(S_1, \phi)) \vee (\neg\, C \wedge \mathtt{wp}(S_2, \phi)) \\
\mathtt{wp}(S_1; S_2, \phi) & \rightsquigarrow & \mathtt{wp}(S_1, \mathtt{wp}(S_2, \phi)) \\
\mathtt{wp}(x = e, \phi) & \rightsquigarrow & \phi[x \leftarrow e] \\
\mathtt{wp}(\mathtt{error}, \phi) & \rightsquigarrow & \bot \\
\mathtt{wp}(\mathtt{while}\ C\ B, \phi) & \rightsquigarrow & \mathcal{I}(\overline{x}) \wedge \\
 & & \forall \overline{x}, \kappa((\mathcal{I}(\overline{x}) \wedge C \wedge \kappa = \mathcal{B}(\overline{x}) \rightarrow \mathtt{wp}(B, \mathcal{I}(\overline{x}) \wedge \kappa \prec \mathcal{B}(\overline{x}))) \wedge \\
 & & (\mathcal{I}(\overline{x}) \wedge \neg\, C \rightarrow \phi)) \\
\mathtt{wp}(\mathtt{return}, \phi) & \rightsquigarrow & \phi \\
\mathtt{wp}(f(\bar{i}, \overline{o}), \phi) & \rightsquigarrow & \mathcal{S}_f(\bar{i}, \overline{o}) \rightarrow \phi
\end{array}
$$

Fig. 1: Weakest precondition calculus rules for a simple imperative language.

predicate transformer. A well-known transformer is the weakest precondition of P with respect to a formula ϕ denoted by $\mathsf{wp}(P, \phi)$. Formally, $\mathsf{wp}(P, \phi)$ is the weakest condition that needs to hold before executing P such that the execution terminates and the post-condition ϕ holds at the end of the execution. Informally, a Hoare triple $\{Pre\}P\{Post\}$ is valid if and only if $Pre \rightarrow \mathsf{wp}(P, Post)$. The rules that define the wp transformer are shown in Figure 1. The symbol $\mathcal{I}$ denotes a loop invariant and $\mathcal{B}$ denotes a loop variant. The symbol $\mathcal{S}$ denotes the function summary and $\phi[x \leftarrow e]$ represents the formula obtained by syntactically replacing all occurrences of x by e. The symbol $\prec$ is a well-founded relation, i.e., it does not admit any infinite chain.

2.3. From Weakest preconditions calculus to CHCs

We can obtain a set of CHCs by first applying exhaustively the rules in Figure 1 to the formula:

$$Pre \rightarrow \mathsf{wp}(P, Post) \wedge \bigwedge_{f \in P} \forall \bar{i}, \overline{o}.\mathsf{wp}(B_f, S_f(\bar{i}, \overline{o}))$$

where B_f is the body of the function f. While the result is not syntactically CHC, it can be put into the syntactically correct form by applying *negation normal form*, *prenex normal form*, and finally *conjunctive normal form* transformations[2]. Finally, we can use many of the abstract interpretation and SMT-based model checking CHC solvers [Komuravelli et al. 2013; McMillan and Rybalchenko 2013; Hoder and Bjørner 2012b; Bjørner et al. 2013; Gange et al. 2013; Hermenegildo et al. 2003; Henriksen and Gallagher 2006; Rümmer et al. 2013] for inferring the unknown relations $\mathcal{I}, \mathcal{B}, \mathcal{S}$.

To illustrate, Figure 2(a) shows a program which adds two numbers. We would like establish validity of the Hoare triple $\{y \geq 0\}P\{x = x_{old} + y_{old}\}$, where P encodes lines 1–6. Figure 2(b) shows the corresponding verification conditions obtained after applying exhaustively the weakest preconditions calculus rules from Figure 1. Note that the VCs are expressed as Constrainted Horn Clauses. The relation pre represents the preconditions of the program. The relation $\mathcal{I}$ expresses the loop invariant which we must infer in order to prove our Hoare triple. The relation exit represents the state after the loop exit is executed, and finally, the relation error expresses our error condition. The clauses C_3, C_4, and C_5 are originated from the rule for `while`. Clause C_1 represents the preconditions of our program and clause C_2 is originated from the rule for sequential composition. Finally, for proving our postcondition we actually generate the following code `if` $(x \neq x_{old} + y_{old})$ `error`. This is the reason why clause C_6 describes our postcondition in negated form.

[2]Note that the variant $\mathcal{B}$ is a function. Thus, the result is non-CHC. In practice, $\mathcal{B}$ is dropped for safety or reachability properties, and turned into a well-founded relation for termination properties.

$$
\begin{array}{l}
\{\ \text{Pre: } y \geq 0\ \} \\
\langle 1 \rangle x_{old} = x; \\
\langle 2 \rangle y_{old} = y; \\
\langle 3 \rangle \textbf{while}\ (y > 0)\ \{ \\
\langle 4 \rangle \quad x = x + 1; \\
\langle 5 \rangle \quad y = y - 1; \\
\langle 6 \rangle \} \\
\{\ \text{Post: } x = x_{old} + y_{old}\ \}
\end{array}
$$

(a)

$$
\begin{array}{llcl}
C_1: & \mathsf{pre}(x, y) & \leftarrow & y \geq 0\ . \\
C_2: & \mathcal{I}(x, y, x_{old}, y_{old}) & \leftarrow & \mathsf{pre}(x, y), \\
 & & & x_{old} = x,\ y_{old} = y\ . \\
C_3: & \mathcal{I}(x, y, x_{old}, y_{old}). & & \\
C_4: & \mathcal{I}(x', y', x_{old}, y_{old}) & \leftarrow & \mathcal{I}(x, y, x_{old}, y_{old}) \\
 & & & y > 0, \\
 & & & x' = x + 1, \\
 & & & y' = y - 1. \\
C_5: & \mathsf{exit}(x, x_{old}, y_{old}) & \leftarrow & \mathcal{I}(x, y, x_{old}, y_{old}), \\
 & & & y \leq 0. \\
C_6: & \mathsf{error}(x, x_{old}, y_{old}) & \leftarrow & x \neq x_{old} + y_{old} \\
C_7: & \bot & \leftarrow & \mathsf{error}(x, x_{old}, y_{old})
\end{array}
$$

(b)

Fig. 2: Program and its Verification Conditions encoded as CHCs.

The Hoare triple $\{y \geq 0\}P\{x = x_{old} + y_{old}\}$ holds if the query C_7 is satisfiable. If we solve this query together with clauses $C_1, \ldots, C_6$ using SPACER [Komuravelli et al. 2014], we obtain the safe inductive invariant :

$$\mathcal{I}(x, y, x_{old}, y_{old}) \leftrightarrow x = x_{old} - y + y_{old} \wedge y \geq 0$$

3. SEAHORN

In this section, we describe SEAHORN, a concrete example of an algorithmic logic-based verification framework. SEAHORN is a fully automated verifier that proves user-supplied assertions as well as a number of built-in safety properties. For example, SEAHORN provides built-in checks for buffer and signed integer overflows. It is released as open-source and its source code is publicly available at http://tinyurl.com/GetSeaHorn.

3.1. Design and implementation

The design of SEAHORN provides users, developers, and researchers with an extensible and customizable environment for experimenting with and implementing new software verification techniques. It has been developed in a modular fashion, inspired by the design of modern compilers. SEAHORN overall architecture is illustrated in Figure 3. Its architecture is layered in three parts:

— **Front-End**: Takes an LLVM-based (e.g., C) input program and generates LLVM IR bitcode. Specifically, it performs the pre-processing and optimization of the bitcode for verification purposes.
— **Middle-end**: Takes as input the optimized LLVM bitcode and emits verification condition as CHC. The middle-end is in charge of selecting encoding of the VCs and the degree of precision.
— **Back-End**: Takes CHC as input and outputs the result of the analysis. In principle, any verification engine that digests CHC clauses could be used to discharge the VCs. Currently, SEAHORN employs several SMT-based model checking engines based on PDR/IC3 [Bradley 2012], including SPACER [Komuravelli et al. 2013; Komuravelli et al. 2014] and GPDR [Hoder and Bjørner 2012b]. Complementary, SEAHORN uses the abstract interpretation-based analyzer IKOS (Inference Kernel for Open Static Analyzers) [Brat et al. 2014] for providing numerical invariants.

This layered architecture allows to separate the concerns of the input language syntax, its operational semantics, and and the underlying verification semantics – the semantics used by the verification engine.

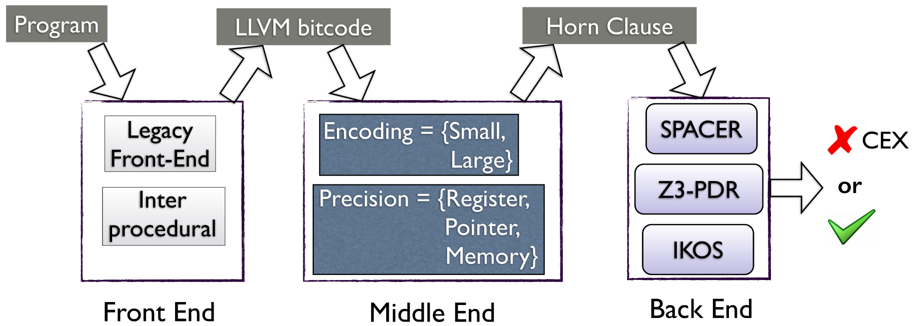


Fig. 3: Overview of SEAHORN architecture.

In the front-end, SEAHORN provides two options: a *legacy* front-end and an *inter-procedural* front-end. The former, has been originally developed for UFO [Albarghouthi et al. 2013], and it has been very effective for solving SV-COMP (2013, 2014, and 2015) problems. However, it has its own limitations: its design is not modular and it relies on multiple unsupported legacy tools (such as `llvm-gcc` and LLVM versions 2.6 and 2.9). Thus, it is difficult to maintain and extend. The inter-procedural front-end, is a generic, modular and easy to maintain front-end. It takes any input program that can be translated into LLVM bitcode. Currently, SEAHORN uses `clang` and `gcc` via `DragonEgg` [3]. In a long run, our goal is to make SEAHORN not to be limited to C programs, but applicable (with various degrees of success) to a broader set of languages based on LLVM (e.g., C++, Objective C, and Swift). The generated LLVM bitcode is then preprocessed and optimized in order to simplify the verification task. Moreover, the inter-procedural front-end provides a transformation based on the concept of *mixed semantics*[4] [Gurfinkel et al. 2008; Lal and Qadeer 2014]. Such transformation, is essential when proving safety of large programs and assertions are nested deeply inside the call graph.

In the middle-end, SEAHORN is fully parametric in the semantics (e.g., small-step, big-step, *etc*) used for the generation of VCs. In addition to generating VCs based on small-step semantics [Peralta et al. 1998], SEAHORN can also automatically lift small-step semantics to large-step [Beyer et al. 2009; Gurfinkel et al. 2011] (*a.k.a.* Large Block Encoding, or LBE). The level of abstraction in the built-in semantics varies from considering only LLVM numeric registers (scalars) to considering the whole heap (modeled as a collection of non-overlapping arrays).

In the back-end, SEAHORN builds on the state-of-the-art in Software Model Checking (SMC) and Abstract Interpretation (AI). SMC and AI have independently led over the years to the production of analysis tools that have a substantial impact on the development of real world software. Interestingly, the two exhibit complementary strengths and weaknesses (see e.g., [Gurfinkel and Chaki 2010; Albarghouthi et al. 2012; Garoche et al. 2013; Bjørner and Gurfinkel 2015]). While SMC so far has been proved stronger on software that is mostly control driven, AI is quite effective on data-dependent programs. SEAHORN combines SMT-based model checking techniques with program invariants supplied by an abstract interpretation-based tool.

[3]`DragonEgg` (http://dragonegg.llvm.org/) is a GCC plugin that replaces GCC's optimizers and code generators with those from LLVM. As result, the output can be LLVM bitcode.
[4]The term *mixed* semantics refers to a combination of small- with big-step operational semantics.

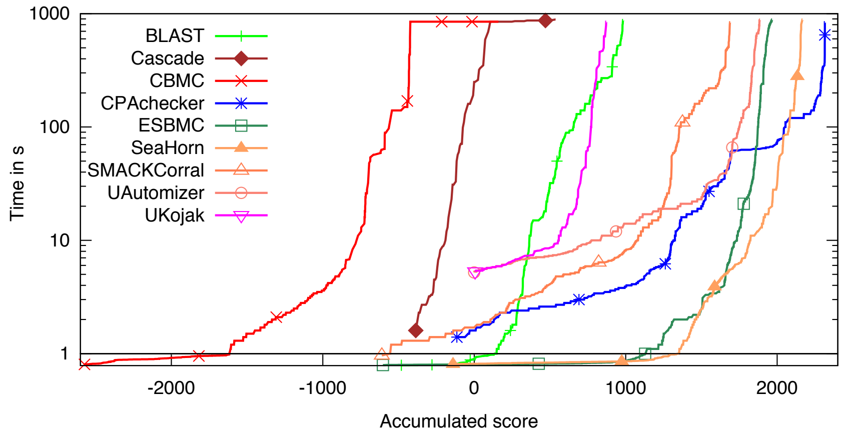


Fig. 4: Quantile graph of the results for the *Control Flow* category.

3.2. Comparative evaluation with other software verifiers

SEAHORN has participated in the International Competition of Software Verification[5] (SV-COMP 2015) [Beyer 2015]. In this competition, SEAHORN the legacy non-inter-procedural front-end. It was configured to use the large step semantics and IKOS with interval abstract domain.

Overall, SEAHORN won one gold medal in the *Simple* category – benchmarks that depend mostly on control-flow structure and integer variables – two silver medals in the categories *Device Drivers* and *Control Flow*. The former is a set of benchmarks derived from the Linux device drivers and includes a variety of C features including pointers. The latter is a set of benchmarks dependent mostly on the control-flow structure and integer variables. In the device drivers category, SEAHORN was beaten only by BLAST [Beyer et al. 2007] – a tool tuned to analyzing Linux device drivers. Specifically, BLAST got 88% of the maximum score while SEAHORN got 85%. The *Control Flow* category, was won by CPAChecker [Beyer and Keremoglu 2011] getting 74% of the maximum score, while SEAHORN got 69%. However, as can be seen in the quantile plot reported in the Figure 4, SEAHORN is significantly more efficient than most other tools solving most benchmarks much faster.

Subsequently, we have tested SEAHORN inter-procedural verification capabilities. We ran several experiments on the 215 benchmarks that we either could not verify or took more than a minute to verify in SV-COMP 2015. For example, we compared the running times with and without inlining in the front-end. Figure 5 shows a scatter plot of the running times and we see that SPACER takes less time on many benchmarks when inlining is disabled.

3.3. Evaluation on an industrial case-study

We also evaluated the SEAHORN built-in buffer overflow checks on two autopilot control software. We have used two open-source autopilot control software mnav[6] (160K LOC) and paparazzi[7] (20K LOC). Both are versatile autopilot control software for a fixed-wing aircrafts and multi-copters. Overall, SEAHORN was able to prove the ab-

[5]Detailed results can be found at http://tinyurl.com/svcomp15
[6]Micro NAV Autopilot Software available at http://sourceforge.net/projects/micronav/.
[7]Paparazzi Autopilot Software available at http://wiki.paparazziuav.org/wiki/Main_Page.

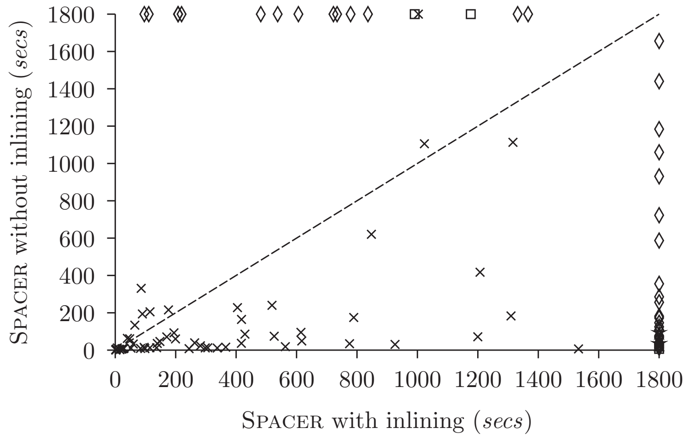


Fig. 5: A plot showing the advantage of inter-procedural (y-axis) versus intra-procedural (x-axis) encodings using SPACER back-end.

sence of buffer overflows for both benchmarks. To the best of our knowledge, this is the first time that absence of buffer overflows has been proven for mnav.

4. CONCLUSIONS

Developing new tools for automated software verification is a tedious and very difficult task. First, due to the undecidability of the problem tools must be highly tuned and engineered to provide reasonable efficiency/precision trade-offs. Second, there is a very diverse assortments of syntactic and semantic features in the different programming languages. In this article, we advocate for a design that allows the decoupling of programming language syntax and semantics from the underlying verification technique. We claim that *Constrained Horn Clauses (CHCs)* is the ideal candidate to be the intermediate formal language for software verification. CHCs are a uniform way to formally represent transition systems while allowing many different encoding styles of verification conditions. This is inline with recent trends in the software verification community and advocated by Bjørner et al. [Bjørner et al. 2012].

We also presented, SEAHORN, an LLVM-based automated verification framework. By its very nature, a verifier shares many of the complexities of an optimizing compiler and of an efficient automated theorem prover. From the compiler perspective, the issues include idiomatic syntax, parsing, intermediate representation, static analysis, and equivalence preserving program transformations. From the theorem proving perspective, the issues include verification logic, verification condition generation, synthesizes of sufficient inductive invariants, deciding satisfiability, interpolation, and consequence generation. By reducing verification to satisfiability of CHC, SEAHORN cleanly separates between compilation and verification concerns and lets us re-use many of the existing components (from LLVM and Z3). SEAHORN is a versatile and highly customizable framework that helps significantly in building new tools by allowing researchers to experiment only on their particular techniques of interest. We have shown that SEAHORN is a highly competitive verifier for safety properties both for verification benchmarks (SV-COMP) and large industrial software (autopilot code).

This is an exciting time for algorithmic software verification. The advances in the computational capabilities of hardware and maturity of verification algorithms make the technology scalable, accessible, and applicable to serious industrial applications. We believe that the line of work presented in this article provides the necessary foundations for building the next-generation verification tools, and will facilitate simpler designs and better communication of verification results between tools and their users.

5. ACKNOWLEDGMENTS

We would like to thank Anvesh Komuravelli and Nikolaj Bjørner for numerous discussions that helped shape this work.

REFERENCES

Aws Albarghouthi, Arie Gurfinkel, and Marsha Chechik. 2012. Craig Interpretation. In *SAS*. 300–316.

Aws Albarghouthi, Arie Gurfinkel, Yi Li, Sagar Chaki, and Marsha Chechik. 2013. UFO: Verification with Interpolants and Abstract Interpretation - (Competition Contribution). In *TACAS*. 637–640.

Aws Albarghouthi, Yi Li, Arie Gurfinkel, and Marsha Chechik. 2012. Ufo: A Framework for Abstraction- and Interpolation-Based Software Verification. In *CAV*. 672–678.

Dirk Beyer. 2015. Software Verification and Verifiable Witnesses (Report on SV-COMP 2015). In *TACAS*.

Dirk Beyer, Alessandro Cimatti, Alberto Griggio, M. Erkan Keremoglu, and Roberto Sebastiani. 2009. Software model checking via large-block encoding. In *FMCAD*. 25–32.

Dirk Beyer, Thomas A. Henzinger, Ranjit Jhala, and Rupak Majumdar. 2007. The software model checker Blast. *STTT* 9, 5-6 (2007), 505–525.

Dirk Beyer and M. Erkan Keremoglu. 2011. CPAchecker: A Tool for Configurable Software Verification. In *CAV*. 184–190.

Nikolaj Bjørner and Arie Gurfinkel. 2015. Property Directed Polyhedral Abstraction. In *Verification, Model Checking, and Abstract Interpretation - 16th International Conference, VMCAI 2015*. 263–281.

Nikolaj Bjørner, Kenneth L. McMillan, and Andrey Rybalchenko. 2012. Program Verification as Satisfiability Modulo Theories. In *SMT*. 3–11.

Nikolaj Bjørner, Kenneth L. McMillan, and Andrey Rybalchenko. 2013. On Solving Universally Quantified Horn Clauses. In *SAS*. 105–125.

Aaron R. Bradley. 2012. IC3 and beyond: Incremental, Inductive Verification. In *CAV*. 4.

Guillaume Brat, Jorge A. Navas, Nija Shi, and Arnaud Venet. 2014. IKOS: A Framework for Static Analysis Based on Abstract Interpretation. In *SEFM*. 271–277.

Maurice Bruynooghe, Gerda Janssens, Alain Callebaut, and Bart Demoen. 1987. Abstract Interpretation: Towards the Global Optimization of Prolog Programs. In *Proceedings of the 1987 Symposium on Logic Programming, San Francisco, California, USA, August 31 - September 4, 1987*. 192–204.

Edmund M. Clarke and E. Allen Emerson. 1981. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *Logics of Programs, Workshop, Yorktown Heights, New York, May 1981*. 52–71.

Patrick Cousot and Radhia Cousot. 1977. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In *Proceedings of the Fourth Annual Symposium on Principles of Programming Languages*. 238–252.

Emanuele De Angelis, Fabio Fioravanti, Alberto Pettorossi, and Maurizio Proietti. 2014. VeriMAP: A Tool for Verifying Programs through Transformations. In *TACAS*. 568–574.

Giorgio Delzanno and Andreas Podelski. 1999. Model Checking in CLP. In *TACAS*. 223–239.

Edsger W. Dijkstra. 1975. Guarded Commands, Nondeterminacy and Formal Derivation of Programs. *Commun. ACM* 18, 8 (1975), 453–457.

Cormac Flanagan. 2003. Automatic Software Model Checking Using CLP. In *ESOP*. 189–203.

Robert W. Floyd. 1967. Assigning meanings to programs. *Symposium Applied Mathematics* 10 (1967), 19–32.

Graeme Gange, Jorge A. Navas, Peter Schachte, Harald Søndergaard, and Peter J. Stuckey. 2013. Failure tabled constraint logic programming by interpolation. *TPLP* 13, 4-5 (2013), 593–607.

Pierre-Loïc Garoche, Temesghen Kahsai, and Cesare Tinelli. 2013. Incremental Invariant Generation Using Logic-Based Automatic Abstract Transformers. In *NASA Formal Methods, 5th International Symposium, NFM 2013*. 139–154.

Sergey Grebenshchikov, Nuno P. Lopes, Corneliu Popeea, and Andrey Rybalchenko. 2012. Synthesizing Software Verifiers from Proof Rules. In *PLDI*. 405–416.

Ashutosh Gupta, Corneliu Popeea, and Andrey Rybalchenko. 2011. Threader: A Constraint-Based Verifier for Multi-threaded Programs. In *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*. 412–417.

Arie Gurfinkel and Sagar Chaki. 2010. Combining predicate and numeric abstraction for software model checking. *STTT* 12, 6 (2010), 409–427.

Arie Gurfinkel, Sagar Chaki, and Samir Sapra. 2011. Efficient Predicate Abstraction of Program Summaries. In *NFM*. 131–145.

Arie Gurfinkel, Temesghen Kahsai, and Jorge A. Navas. 2015. SeaHorn: A Framework For Verifying C Programs - (Competition Contribution). In *To appear in TACAS*.

Arie Gurfinkel, Ou Wei, and Marsha Chechik. 2008. Model Checking Recursive Programs with Exact Predicate Abstraction. In *ATVA*. 95–110.

Kim S. Henriksen and John P. Gallagher. 2006. CHA: Convex Hull Analyser for constraint logic programs. (2006).

Manuel V. Hermenegildo, Germán Puebla, Francisco Bueno, and Pedro López-García. 2003. Program Development Using Abstract Interpretation (And The Ciao System Preprocessor). In *SAS*. 127–152.

C. A. R. Hoare. 1969. An Axiomatic Basis for Computer Programming. *Commun. ACM* 12, 10 (1969), 576–580.

Krystof Hoder and Nikolaj Bjørner. 2012a. Generalized Property Directed Reachability. In *Theory and Applications of Satisfiability Testing - SAT 2012 - 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings*. 157–171.

Krystof Hoder and Nikolaj Bjørner. 2012b. Generalized Property Directed Reachability. In *SAT*. 157–171.

Joxan Jaffar and Jean-Louis Lassez. 1987. Constraint Logic Programming. In *POPL*. 111–119.

Joxan Jaffar, Vijayaraghavan Murali, Jorge A. Navas, and Andrew E. Santosa. 2012. TRACER: A Symbolic Execution Tool for Verification. In *CAV*. 758–766.

Joxan Jaffar, Andrew E. Santosa, and Razvan Voicu. 2004. A CLP Proof Method for Timed Automata. In *RTSS*. 175–186.

James C. King. 1976. Symbolic Execution and Program Testing. *Commun. ACM* 19, 7 (1976), 385–394.

Anvesh Komuravelli, Arie Gurfinkel, and Sagar Chaki. 2014. SMT-Based Model Checking for Recursive Programs. In *CAV*. 17–34.

Anvesh Komuravelli, Arie Gurfinkel, Sagar Chaki, and Edmund M. Clarke. 2013. Automatic Abstraction in SMT-Based Unbounded Software Model Checking. In *CAV*. 846–862.

Akash Lal and Shaz Qadeer. 2014. A program transformation for faster goal-directed search. In *FMCAD*. 147–154.

Chris Lattner and Vikram S. Adve. 2004. LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation. In *CGO*. 75–88.

Kim. Marriott and Peter. J. Stuckey. 1998. *Introduction to Constraint Logic Programming*. MIT Press, Cambridge, MA, USA.

John McCarthy. 1963. A Basis for a Mathematical Theory of Computation. (1963), 33–70.

Ken McMillan and Andrey Rybalchenko. 2013. *Solving Constrained Horn Clauses using Interpolation*. Technical Report. MSR-TR-2013-6.

Mario Méndez-Lojo, Jorge A. Navas, and Manuel V. Hermenegildo. 2007. A Flexible, (C)LP-Based Approach to the Analysis of Object-Oriented Programs. In *LOPSTR*. 154–168.

Kalyan Muthukumar and Manuel V. Hermenegildo. 1989. Determination of Variable Dependence Information through Abstract Interpretation. In *Logic Programming, Proceedings of the North American Conference*. 166–185.

Peter Naur. 1966. Proof of algorithms by general snapshots. 6 (1966), 310–316. Issue 4.

Julio C. Peralta, John P. Gallagher, and Hüseyin Saglam. 1998. Analysis of Imperative Programs through Analysis of Constraint Logic Programs. In *SAS*. 246–261.

Jean-Pierre Queille and Joseph Sifakis. 1982. Specification and verification of concurrent systems in CESAR. In *International Symposium on Programming, 5th Colloquium, Torino, Italy, April 6-8, 1982, Proceedings*. 337–351.

Philipp Rümmer, Hossein Hojjat, and Viktor Kuncak. 2013. Disjunctive Interpolants for Horn-Clause Verification. In *CAV*. 347–363.

Harald Søndergaard. 1986. An Application of Abstract Interpretation of Logic Programs: Occur Check Reduction. In *ESOP*. 327–338.

Alan Turing. 1949. Checking a Large Routine. (1949).

Richard Warren, Manuel V. Hermenegildo, and Saumya K. Debray. 1988. On the Practicality of Global Flow Analysis of Logic Programs. In *ICLP*. 684–699.

BULLETIN

SIGLOG Monthly 169
April 1, 2015

****
* Past issues of the newsletter are available at
 http://lii.rwth-aachen.de/lics/newsletters/
* Instructions for submitting an announcement to the newsletter
 can be found at
 http://lii.rwth-aachen.de/lics/newsletters/inst.html
****

TABLE OF CONTENTS

ICTAC 2015 - Call for Papers
SETTA 2015 - Call for Papers
HaPoC 3 - Call for Papers
WoLLIC - Call for Participation
GBMSC - Call for Participation
GBMSC THEME 2 - Call for Participation
ABZ 2016 - Call for Papers, Answers to the case study, Workshops, Tutorials
* JOB ANNOUNCEMENTS
PhD Studentship in Algorithms and Complexity at Royal Holloway, U. of London
New Doctoral Program on Logical Methods in Computer Science (LogiCS), Vienna

SIGLOG MONTHLY BULLETIN
* From the new year, the preparation of the LICS Newsletter is overseen by SIGLOG and its name is changing to SIGLOG Monthly Bulletin. The newsletter will be distributed monthly in the same way as before. Conference announcements should be sent to las-lics@lists.tu-berlin.de following the instructions at http://lics.rwth-aachen.de/newsletters/inst.html

LICS'15 PREVIEW
http://lics.rwth-aachen.de/lics15/
* LICS'15 will colocate with ICALP 2015, 6-10 July, in Kyoto. Masahito Hasegawa (RIMS, Kyoto) is the Conference Chair. The PC is chaired by Catuscia Palamidessi. There will be 4 invited talks, 2 of which joint with ICALP Track B. The 2 invited tutorial talks of 1.5 hours each will be embedded in the main programme.

ACM SIGLOG ANNOUNCEMENT
http://siglog.acm.org
* The ACM has recently chartered a Special Interest Group on Logic and Computation (ACM SIGLOG). Its first Chair is Prakash Panangaden, the other officers are Luke Ong (vice-Chair), Natarajan Shankar (Treasurer) and Alexandra Silva (Secretary).
* The ACM-IEEE Symposium on Logic in Computer Science will be the flagship conference of SIGLOG. SIGLOG will also actively seek association agreements with other conferences in the field. A SIGLOG newsletter will be published quarterly in an electronic format with community news, technical columns, members' feedback, conference reports, book reviews and other items of interest to the community.
* One can join SIGLOG by visiting
https://campus.acm.org/public/qj/gensigqj/siglist/gensigqj˙siglist.cfm
It is possible to join SIGLOG without joining ACM (the SIGLOG membership fee is $25 and $15 for students).

DATES
* TACL 2015
Deadline for early registration (conference): April 30, 2015
Deadline for registration (school): April 30, 2015
School dates: June 15-19, 2015, University of Salerno (Italy)

Conference dates: June 21-26, 2015, Ischia Island (Italy)
http://logica.dmi.unisa.it/tacl/
* NLCS '15
Paper submission deadline: April 2, 2015
http://www.indiana.edu/˜iulg/nlcs.html
* CSL 2015
Abstract submission: April 3, 2015
Paper Submission: April 10, 2015
http://logic.las.tu-berlin.de/csl2015/
* MFPS XXXI
Submission deadline: April 3, 2015
http://events.cs.bham.ac.uk/mfps31/
* MeMo 2015
Submission deadline: April 4, 2015
https://discotec2015.inria.fr/memo-2015/
* WISTP 2015
Paper submission: April 10, 2015
http://www.wistp.org
* LOGIC AT NATAL 2015
Submission deadline: April 12, 2015
http://natalogic-2015.dimap.ufrn.br/
* CONCUR 2015
Submission of Abstracts: April 13th, 2015
Submission of Papers: April 20th, 2015 (firm)
http://mafalda.fdi.ucm.es/concur2015
* ACKERMANN AWARD 2015
Nomination deadline: April 15, 2015
* HDRA 2015
Submission deadline: April 15, 2015
http://rdp15.mimuw.edu.pl/index.php?site=hdra
* CADE-25 POSTERS & ENCYCLOPEDIA ENTRIES
Submission deadline: April 19, 2015
http://proofsystem.github.io/Encyclopedia/
* LPNMR 2015
Paper registration: April 19th, 2015
Paper submission: April 24th, 2015
http://lpnmr2015.mat.unical.it/
* FORMATS 2015
Abstract submission: April 20, 2015
Paper submission: April 27, 2015
http://formats2015.unifi.it
* ICLP 2015
Abstracts due: April 20, 2015
Papers due: April 27, 2015
http://booleconferences.ucc.ie/iclp2015
* ATVA 2015
Abstract submission: April 22, 2015
Paper submission: April 25, 2015
Workshop proposal submission: May 5, 2015
http://atva2015.ios.ac.cn/
* FCT 2015
Submission deadline: April 25, 2015

http://sites.google.com/site/fct2015gdansk/
* WL4AI 2015 (IJCAI-2015 Workshop)
Submission deadline: April 27, 2015
http://www.iiia.csic.es/wl4ai-2015/
* NFM 2015
Conference: April 27-29, 2015
http://www.NASAFormalMethods.org/nfm2015
* E. W. BETH DISSERTATION PRIZE 2015
Deadline for Submissions: April 27th, 2015.
http://www.folli.info/?page˙id=84
* LFMTP 2015
Abstract submission: April 30, 2015
Paper submission: May 7, 2015
http://lfmtp.org/2015
* TIME 2015
Submission deadline: April 30, 2015
http://time2015.uni-kassel.de
* DOMAINS XII
Abstract submission: May 1, 2015
Conference: August 25-28, 2015
http://booleconferences.ucc.ie/gbmsc2015/domainsxii
* QPL 2015
Submission deadline: 1 May 2015
http://www.cs.ox.ac.uk/qpl2015
* FroCoS 2015
Abstract submission: May 4, 2015
Full paper submission: May 10, 2015
http://frocos2015.ii.uni.wroc.pl/
* AUTOMATHA 2015
Conference: May 6-9, 2015
http://www.automatha.uni-leipzig.de
* QUANTIFY 2015
Paper submission: May 8, 2015
Workshop: August 1, 2015
http://fmv.jku.at/quantify15/
* SSFT 2013
Summer school: May 17-22, 2015
http://fm.csl.sri.com/SSFT15
* ICTAC 2015
Abstract submission: June 1, 2015
Paper submission: June 5, 2015
Submission deadline: June 5, 2015
http://www.ictac2015.co
* SETTA 2015
Abstract deadline: June 12, 2015
Paper submission: June 19, 2015
http://cs.nju.edu.cn/setta/
* HaPoC 3
Submission deadline: June 19, 2015
http://hapoc2015.di.unipi.it
* WoLLIC 2015
Workshop: July 20-23, 2015

http://www.indiana.edu/~iulg/wollic/
* GBMSC
Conference: August 17-28, 2015
http://booleconferences.ucc.ie/gbmsc2015
* GBMSC THEME 2
Conference: August 27-28, 2015
http://booleconferences.ucc.ie/gbmsc2015/theme2
* ABZ 2016
Workshop proposal submissions: October 16, 2015
Workshop proposal notifications: November 6, 2015
Research paper and answers to case study submission: January 15, 2016
Short paper submission: February 4, 2016
Tutorial proposal submissions: February 15, 2016
http://www.cdcc.faw.jku.at/ABZ2016/

TACL2015 - TOPOLOGY, ALGEBRA, AND CATEGORIES IN LOGIC
Call for Conference Submissions and School Participation
School: 15 - 19 June 2015, University of Salerno (Italy)
Conference: 21 - 26 June 2015, Ischia Island (Italy)
http://logica.dmi.unisa.it/tacl/
* AIMS
The programme of the conference TACL 2015 will focus on three interconnecting mathematical themes central to the semantic study of logics and their applications: algebraic, categorical, and topological methods. This year the school will be held at the campus of the University of Salerno and will include four tutorials, each consisting of 1.5 hour lectures for five days.
* IMPORTANT DATES
Deadline for early registration (conference) 30 April 2015
Deadline for registration (school) 30 April 2015
* INVITED SPEAKERS
Olivia Caramello (Institut des Hautes Etudes Scientifiques), Agata Ciabattoni (Technische Universitaet Wien), Maria Manuel Clementino (Universidade de Coimbra), Emil Jerabek (Academy of Sciences of the Czech Republic), Andre Joyal (Universite du Quebec), Keith A. Kearnes (University of Colorado), Daniele Mundici (University of Florence), Paulo Oliva (Queen Mary University of London), Jorge Picado (Universidade de Coimbra), Michael Pinsker (University Paris Diderot)
* SCHOOL LECTURERS
Guram Bezhanishvili (New Mexico State University), Brian Davey (La Trobe University), Ieke Moerdijk (Nijmegen University), Luke Ong (Oxford University)

3RD WORKSHOP ON NATURAL LANGUAGE AND COMPUTER SCIENCE (NLCS'15)
July 5, 2015
Kyoto, Japan
http://www.indiana.edu/~iulg/nlcs.html
Affiliated with ICALP 2015 & LICS 2015
Endorsed by the Association for Computational Linguistics Special Interest Group on Computational Semantics
* AIMS AND SCOPE

Formal tools coming from logic and category theory are important in both natural language semantics and in computational semantics. Moreover, work on these tools borrows heavily from all areas of theoretical computer science. In the other direction, applications having to do with natural language has inspired developments on the formal side. The workshop invites papers on both topics. Specific topics includes, but are not limited to:

- logic for semantics of lexical items, sentences, discourse and dialog
- continuations in natural language semantics
- formal tools in textual inference, such as logics for natural language inference
- applications of category theory in semantics
- linear logic in semantics
- formal approaches to unifying data-driven and declarative approaches to semantics

* INVITED SPEAKERS

Makoto Kanazawa, National Institute of Informatics
Gerard de Melo, Tsinghua University

* PAPER SUBMISSIONS

Extended abstracts of up to 10 pages may be submitted through Easychair:
https://easychair.org/conferences/?conf=nlcs15

* ORGANIZERS

Valeria de Paiva
Makoto Kanazawa
Larry Moss

* IMPORTANT DATES

Paper submission deadline: April 2, 2015
Author notification: May 1, 2015
Electronic versions of papers due: May 14, 2015
Workshop: July 5, 2015

COMPUTER SCIENCE LOGIC 2015 (CSL 2015)

Call for Papers
7-10 September 2015
Berlin, Germany
http://logic.las.tu-berlin.de/csl2015/

* AIM AND SCOPE

Computer Science Logic (CSL) is the annual conference of the European Association for Computer Science Logic (EACSL). The conference is intended for computer scientists whose research activities involve logic, as well as for logicians working on issues significant for computer science.

* LOCATION

The 24th EACSL Annual Conference on Computer Science Logic will be held at the Technical University Berlin from Monday, 7 September 2015 to Thurday, 10 September 2015.

* LIST OF TOPICS OF INTEREST (NON EXHAUSTIVE)

- automated deduction and interactive theorem proving
- constructive mathematics and type theory
- equational logic and term rewriting
- automata and games, game semantics
- modal and temporal logic

- model checking
- decision procedures
- logical aspects of computational complexity
- finite model theory
- computational proof theory
- bounded arithmetic and propositional proof complexity
- logic programming and constraints
- lambda calculus and combinatory logic
- domain theory
- categorical logic and topological semantics
- database theory
- specification, extraction and transformation of programs
- logical aspects of quantum computing
- logical foundations of programming paradigms
- verification and program analysis
- linear logic
- higher-order logic
- nonmonotonic reasoning

* IMPORTANT DATES

Abstract submission: 3 April 2015
Paper Submission: 10 April 2015
Paper Notification: 13 June 2015
Conference: 7 - 10 September 2015

* INVITED SPEAKERS

- Martn Abadi (Google and Santa Cruz)
- Elham Kashefi (Edinburgh)
- Markus Lohrey (Siegen)
- Ryan Williams (Stanford)

* SUBMISSION

Authors are invited to submit papers of not more than 15 pages in LIPIcs style presenting work not previously published. Papers are to be submitted through Easychair. Submitted papers must be in English and must provide sufficient detail to allow the PC to assess the merits of the paper. Full proofs may appear in a technical appendix which will be read at the reviewers' discretion. Authors are strongly encouraged to include a well written introduction which is directed at all members of the program committee.

* SATELLITE EVENTS

- The 11th International Workshop on Fixed Points in Computer Science (FICS'15) will be held on 11 and 12 September 2015 as a co-located event of CSL'15.
- YuriFest: we will celebrate Yuri Gurevich's 75th birthday with a symposium in his honour on 11 September 2015 as a co-located event of CSL'15.
- The annual meeting of the GI Fachgruppe Logik will be organised at the Technical University Berlin in conjunction with CSL'15.

* PC CHAIR

Stephan Kreutzer (Technical University Berlin, pc chair)

* ORGANISING COMMITTEE:

- Christoph Dittmann (Technical University Berlin)
- Viktor Engelmann (Technical University Berlin)
- Stephan Kreutzer (Technical University Berlin, Chair)
- Jana Pilz (Technical University Berlin)

- Roman Rabinovich (Technical University Berlin)
- Sebastian Siebertz (Technical University Berlin)

MATHEMATICAL FOUNDATIONS OF PROGRAMMING SEMANTICS (MFPS XXXI)
Call for Papers
22-25 June 2015, Nijmegen, Netherlands
http://events.cs.bham.ac.uk/mfps31/

* MFPS SERIES
MFPS conferences are dedicated to the areas of mathematics, logic, and computer science that are related to models of computation in general, and to semantics of programming languages in particular. This is a forum where researchers in mathematics and computer science can meet and exchange ideas. The participation of researchers in neighbouring areas is strongly encouraged. This edition of MFPS will be co-located with CALCO.
* IMPORTANT DATES
Submission: April 3, 2015
Notification: May 15, 2015
Final version: May 29, 2015
* INVITED SPEAKERS
Andrew Pitts, Thierry Coquand, Paul B. Levy, Guy McCusker, Sam Staton
* INVITED TUTORIAL SPEAKERS
Matija Pretnar, Daniela Petrisan, Andrzej Murawski
* SPECIAL SESSIONS
algebraic effects, game semantics, homotopy type theory, quantitative semantics
* MORE INFO
For more information please consult the web page.

2ND INTERNATIONAL WORKSHOP ON META MODELS FOR PROCESS LANGUAGES (MeMo 2015)
Call for Papers
June 5, 2015
Grenoble, France
Satellite workshop of DisCoTec 2015
https://discotec2015.inria.fr/memo-2015/

* AIMS
The MeMo workshops aim to bring together researchers and practitioners working on and with metamodels, i.e. framework theories which provide general, structural results simplifying and driving the development of models of systems and languages.
* TOPICS
Topics include (but are not limited to): algebraic/bialgebraic specifications, bigraphical reactive systems, coalgebras, psi-calculus, concurrent logical frameworks, enriched Lawvere theories, mathematical operational semantics, metalanguages for deductive systems, monads, SOS formats, term and graph transformation systems, tile models, ULTraS and FuTS.
* IMPORTANT DATES
4 April 2015 AoE: Submission of full papers and tool presentations (NEW!)
28 April 2015: Notification to authors (papers & tool presentations)
28 April 2015: Submission of posters

5 May 2015: Submission of camera-ready for pre-proceedings
5 May 2015: Notification of posters to authors
5 June 2015: MeMo in Grenoble
* MORE
Detailed information can be found on the webpage.

9TH WISTP INTERNATIONAL CONFERENCE ON INFORMATION SECURITY THEORY AND PRACTICE
Call for Papers
August 24-25, 2015
Crete, Greece
http://www.wistp.org
* Future ICT technologies, such as the concepts of Ambient Intelligence, Cyber-physical Systems, and Internet of Things provide a vision of the Information Society in which: a) people and physical systems are surrounded with intelligent interactive interfaces and objects, and b) environments are capable of recognising and reacting to the presence of different individuals or events in a seamless, unobtrusive, and invisible manner. The success of future ICT technologies will depend on how secure these systems are and to what extent they protect the privacy of individuals and individuals trust them. In 2007, Workshop in Information Security Theory and Practice (WISTP) was created as a forum for bringing together researchers and practitioners in related areas and to encourage interchange and cooperation between the research community and the industrial/consumer community. Based on the growing interest of the participants, 2015 edition is becoming a conference - The 9th WISTP International Conference on Information Security Theory and Practice (WISTP'2015).
* IMPORTANT DATES
- Paper Submission due: April 10, 2015
- Notification to authors: May 30, 2015
- Camera ready due: June 10, 2015
* SPONSOR
Co-Sponsored by IFIP WG 11.2 Pervasive Systems Security.
* Detailed information can be found on the webpage.

LOGIC AT NATAL (NAT@LOGIC 2015)
Call for Papers
Aug 31-Sep 4, 2015
Natal, Brazil
http://natalogic-2015.dimap.ufrn.br/
* NAT@Logic 2015 is a pool of workshops related to Logic in Computer Science, in Philosophy, and in Mathematics.
These include:
+ LSFA X
(10th Workshop on Logical and Semantic Frameworks, with Applications)
+ GeTFun 3.0
(3rd Workshop on Generalizations of Truth-Functionality)
+ Filomena 2
(2nd Workshop on Philosophy, Logic and Applied Metaphysics)
+ LFIs^15

(Workshop commemorating the 15 years of the LFIs)
+ TRS Reasoning School
(TRS = TRS Reasoning School)
The full programme will boast 10 keynote speakers, plus at least 60 contributed talks and 15 tutorials.
* Paper submission: 12 Apr 2015.

26TH INTERNATIONAL CONFERENCE ON CONCURRENCY THEORY (CONCUR 2015)
Call for Papers
1-4 September, 2015
Madrid, Spain
http://mafalda.fdi.ucm.es/concur2015
* GENERAL
CONCUR 2015 will co-locate with the 12th QEST, the 13th FORMATS, the 10th TGC, WS-FM/BEAT, and a collection of workshops, thus producing our "Madrid meet 2015" meeting. The CONCUR 2015 conference chairs are Luca Aceto (Reykjavik University) and David de Frutos-Escrig (Universidad Complutense de Madrid). CONCUR is the main annual meeting devoted to Concurrency Theory.
* IMPORTANT DATES
Submission of Abstracts: April 13th, 2015
Submission of Papers: April 20th, 2015 (firm)
Notification: June 15th, 2015
Final version: July 3rd, 2015
* PROCEEDINGS
For the first time this year CONCUR will have an open publication of its proceedings, initiating a new collection in LIPIcs (Leibniz International Proceedings in Informatics), published under an open access license by Dagstuhl Publishing. Authors will retain full rights over their work. The accepted papers will be published under a CC-BY license.
* INVITED SPEAKERS
Alexandra Silva, Radboud University, The Netherlands
Mohammad Reza Mousavi, Halmstad University, Sweden
James Worrell, University of Oxford, UK
Gianluigi Zavattaro, Universit di Bologna, Italy
* 25TH ANNIVERSARY SPEECH
Matthew Hennessy, Trinity College Dublin, Ireland
* TOPICS
Submissions are solicited in semantics, logics, verification and analysis of concurrent systems. This includes (but is not strictly limited to): Basic models, Logics for concurrency, Models of specialized systems (biology-inspired, hybrid systems, mobile and collaborative, probabilistic, real-time ...), Verification and analysis techniques, Related programming models and Security issues.
* SUBMISSION GUIDELINES
Authors are invited to submit a draft of at most 13 pages including references. The usage of pdflatex and the LIPIcs style file (see http://www.dagstuhl.de/en/publications/lipics/instructions-for-authors/) are mandatory. Submissions will be sent via EasyChair (https://easychair.org/conferences/?conf=concur2015).

ACKERMANN AWARD 2015 - THE EACSL OUTSTANDING DISSERTATION AWARD FOR LOGIC IN COMPUTER SCIENCE

Call for Nominations

* Nominations are now invited for the 2015 Ackermann Award.
PhD dissertations in topics specified by the EACSL and LICS conferences, which were formally accepted as PhD theses at a university or equivalent institution between 1.1.2013 and 31.12.2014 are eligible for nomination for the award. The deadline for submission is 15 April 2015.
* SUBMISSION DETAILS
Nominations can be submitted from 1 January 2015 and should be sent to the chair of the Jury, Anuj Dawar, by e-mail:anuj.dawar@cl.cam.ac.uk
* THE AWARD
The 2015 Ackermann award will be presented to the recipient(s) at the annual conference of the EACSL, 7-10 September 2015, in Berlin (Germany).
The award consists of
- a certificate,
- an invitation to present the thesis at the CSL/LICS conference,
- the publication of the laudatio in the CSL/LICS proceedings,
- travel support to attend the conference, and
- an invitation to present the work to the Kurt Goedel Society in Vienna.

The jury is entitled to give the award to more (or less) than one dissertation in a year.
* JURY
The jury consists of:
- Thierry Coquand (Chalmers University of Gothenburg);
- Anuj Dawar (University of Cambridge), president of EACSL;
- Dexter Kozen (Cornell University), ACM SigLog representative;
- Orna Kupferman (Hebrew University of Jerusalem);
- Daniel Leivant (Indiana University, Bloomington);
- Luke Ong (University of Oxford);
- Jean-Eric Pin (CNRS and University of Paris 7);
- Simona Ronchi Della Rocca (University of Torino), vice-president of EACSL;
* HOW TO SUBMIT
The candidate or his/her supervisor should submit
1. the thesis (ps or pdf file);
2. a detailed description (not longer than 20 pages) of the thesis in ENGLISH (ps or pdf file);
3. a supporting letter by the PhD advisor and two supporting letters by other senior researchers (in English); supporting letters can also be sent directly to Anuj Dawar (anuj.dawar@cl.cam.ac.uk);
4. a short CV of the candidate;
5. a copy of the document asserting that the thesis was accepted as a PhD thesis at a recognized University (or equivalent institution) and that the candidate has received his/her PhD within the specified period.

The submission should be sent by e-mail as attachments to the chairman of the jury, Anuj Dawar: anuj.dawar@cl.cam.ac.uk, with the following subject line and text:
- Subject: Ackermann Award Submission
- Text: Name of candidate, list of attachments

Submission can be sent via several e-mail messages. If this is the case, please indicate it in the text. Letters of support and documents can also be faxed to: Anuj Dawar, Ackermann Award, +44 1223 334678. The Jury has the right to declare submissions to be out of scope or not to meet the requirements.
* The Award is sponsored by the Kurt Goedel Society.

FIFTH WORKSHOP ON HIGHER-DIMENSIONAL REWRITING AND APPLICATIONS (HDRA 2015)
Call for Papers
28-29 June 2015
Warsaw, Poland
http://rdp15.mimuw.edu.pl/index.php?site=hdra
Co-located with the RDP, RTA and TLCA conferences
* HISTORY
Over recent years, rewriting methods have been generalized from strings and terms to richer algebraic structures such as operads, monoidal categories, and more generally higher-dimensional categories. These extensions of rewriting fit in the general scope of higher-dimensional rewriting theory, which has emerged as a unifying algebraic framework. This approach allows one to perform homotopical and homological analysis of rewriting systems (Squier theory). It also provides new computational methods in combinatorial algebra (Artin-Tits monoids, Coxeter and Garside structures), in homotopical and homological algebra (construction of cofibrant replacements, Koszulness property). The workshop is open to all topics concerning higher-dimensional generalizations and applications of rewriting theory.
* INVITED SPEAKERS
John Baez
TBA
* IMPORTANT DATES
- Submission: April 15, 2015
- Notification: May 6, 2015
- Final version: May 20, 2015
- Conference: 28-29 June, 2015
* SUBMITTING
Submissions should consist in an extended abstract, in pdf format, approximatively 5 pages long, in standard article format. The page for uploading those is https://easychair.org/conferences/?conf=hdra2015
* PROCEEDINGS
The accepted extended abstracts will be made available electronically before the workshop.
* ORGANIZERS
- Yves Guiraud (INRIA / Universite Paris 7)
- Philippe Malbos (Universite Claude Bernard Lyon 1)
- Samuel Mimram (Ecole Polytechnique)

CADE-25 POSTER SESSION & TASK-FORCE TOWARDS AN ENCYCLOPEDIA OF PROOF SYSTEMS (EPS)
Call for Posters and Encyclopedia Entries
1st of August 2015
Berlin, Germany

http://proofsystem.github.io/Encyclopedia/

* DESCRIPTION

In this jubilee edition of CADE, we would like to commemorate the multitude of proof systems that form the theoretical foundations for automated deduction. To achieve this goal, this alternative workshop proposes to bring the whole community together in a task-force to produce a concise encyclopedia of proof systems. Every entry in this encyclopedia will follow a given template and will preferably be exactly one page long, displaying the inference rules of the proof system and possibly a few clarifying remarks. The one-page encyclopedia entries will be displayed as posters during CADE (the Conference on Automated Deduction). Submission instructions are available in the website: http://proofsystem.github.io/Encyclopedia/

* IMPORTANT DATES

- Submission: April 19
- Notification: May 15

13TH INTERNATIONAL CONFERENCE ON LOGIC PROGRAMMING AND NON-MONOTONIC REASONING (LPNMR 2015)

Call for Papers
Lexington, KY, USA
September 27-30, 2015
http://lpnmr2015.mat.unical.it/
(Collocated with the 4th Conference on Algorithmic Decision Theory 2015)

* AIMS AND SCOPE

LPNMR 2015 is the thirteenth in the series of international meetings on logic programming and non-monotonic reasoning. LPNMR is a forum for exchanging ideas on declarative logic programming, non-monotonic reasoning, and knowledge representation. The aim of the conference is to facilitate interactions between researchers and practitioners interested in the design and implementation of logic-based programming languages and database systems, and those working in knowledge representation and nonmonotonic reasoning. LPNMR strives to encompass theoretical and experimental studies that have led or will lead to the construction of systems for declarative programming and knowledge representation, as well as their use in practical applications. This edition of LPNMR will feature several workshops, a special session dedicated to the 6th ASP Systems Competition, and will be collocated with the 4th Algorithmic Decision Theory Conference, ADT 2015. Joint LPNMR-ADT Doctoral Consortium will be a part of the program.

Authors are invited to submit papers presenting original and unpublished research on all aspects of non-monotonic approaches in logic programming and knowledge representation. We invite submissions of both long and short papers.

* TOPICS

Conference topics include, but are not limited to:

1. Foundations of LPNMR Systems
2. Implementation of LPNMR systems
3. Applications of LPNMR

* SUBMISSION

LPNMR 2015 welcomes submissions of long papers (13 pages) or short papers

(6 pages) in the following categories:
- Technical papers
- System descriptions
- Application descriptions
The indicated number of pages includes title page, references and figures. All submissions will be peer-reviewed and accepted papers will appear in the conference proceedings published in the Springer-Verlag Lecture Notes in Artificial Intelligence (LNAI/LNCS) series. At least one author of each accepted paper is expected to register for the conference to present the work.
The Program Committee chairs are planning to arrange for the best papers to be published in a special issue of a premiere journal in the field.
LPNMR 2015 will not accept any paper which, at the time of submission, is under review or has already been published or accepted for publication in a journal or another conference. Authors are also required not to submit their papers elsewhere during LPNMR's review period. However, these restrictions do not apply to previous workshops with a limited audience and without archival proceedings.
* ASSOCIATED WORKSHOPS
LPNMR 2015 will include specialized workshops to be held on September 27 prior to the main conference. Currently planned workshops include:
- Grounding, Transforming, and Modularizing Theories with Variables
Organizers: Marc Denecker, Tomi Janhunen
- Action Languages, Process Modeling, and Policy Reasoning
Organizer: Joohyung Lee
- Natural Language Processing and Automated Reasoning
Organizers: Marcello Balduccini, Ekaterina Ovchinnikova, Peter Schueller
- Learning and Nonmonotonic Reasoning
Organizers: Alessandra Russo and Alessandra Mileo
* IMPORTANT DATES (TENTATIVE)
Paper registration: April 19th, 2015
Paper submission: April 24th, 2015
Notification: June 1st, 2015
Final versions due: June 15th, 2015
* VENUE
Lexington is a medium size, pleasant and quiet university town. It is located in the heart of the so-called Bluegrass Region in Central Kentucky. The city is surrounded by beautiful horse farms on green pastures dotted with ponds and traditional architecture stables, and small race tracks, and bordered by white or black fences. The Horse Museum is as beautifully located as it is interesting. Overall, the city has a nice feel that mixes well old and new. The conference will be held in the Hilton Lexington Downtown hotel.
* GENERAL CHAIR
Victor Marek, University of Kentucky, KY, USA
* PROGRAM CHAIRS
Giovambattista Ianni, University of Calabria, Italy
Mirek Truszczynski, University of Kentucky, KY, USA
* WORKSHOPS CHAIR
Yuliya Lierler, University of Nebraska at Omaha, NE, USA
* PUBLICITY CHAIR

Francesco Calimeri, University of Calabria, Italy
* CONTACT
lpnmr2015@mat.unical.it

13TH INTERNATIONAL CONFERENCE ON FORMAL MODELLING AND ANALYSIS OF TIMED SYSTEMS (FORMATS 2015)
Call for Papers
Madrid, Spain
September 2-4, 2015
http://formats2015.unifi.it
* TOPICS
The aim of FORMATS is to promote the study of fundamental and practical aspects of timed systems, and to bring together researchers from different disciplines that share interests in modelling and analysis of timed systems. Typical topics include (but are not limited to):
- Foundations and Semantics: Theoretical foundations of timed systems and languages; comparison between different models (timed automata, timed Petri nets, hybrid automata, timed process algebra, max-plus algebra, probabilistic models).
- Methods and Tools: Techniques, algorithms, data structures, and software tools for analyzing timed systems and resolving temporal constraints (scheduling, worst-case execution time analysis, optimization, model checking, testing, constraint solving, etc.).
- Applications: Adaptation and specialization of timing technology in application domains in which timing plays an important role (real-time software, hardware circuits, and problems of scheduling in manufacturing and telecommunication).
* SUBMISSION
FORMATS 2015 solicits high-quality papers reporting research results and/or experience reports related to the topics mentioned above. Submitted papers must contain original, unpublished contributions, not submitted for publication elsewhere. The papers should be submitted electronically in PDF, following the Springer LNCS style guidelines. Submissions should not exceed 15 pages in length.
* PUBLICATION AND SPECIAL ISSUE
The proceedings of FORMATS 2015 will be published by Springer in the Lecture Notes in Computer Science series.
A special issue dedicated to FORMATS will be hosted in the Journal of Real Time Systems, Springer, collecting the extensions of papers selected by quality and fitness to the Journal scope, and subject to additional revision.
* IMPORTANT DATES
- Abstract submission: April 20, 2015
- Paper submission: April 27, 2015
- Notification of acceptance: June 10, 2015
- Final version due: June 22, 2015
- Conference: September 2-4, 2015
* PROGRAM CHAIRS
Sriram Sankaranarayanan (University of Colorado at Boulder, USA)
Enrico Vicario (University of Florence, Italy)

* COLOCATION
FORMATS 2015 will be colocated with QEST, CONCUR, TGC, EPEW,
and WS-FM/BEAT, EXPRESS/SOS, PV, TRENDS, YR-CONCUR and FOCLASA as part of
Madrid Meet 2015 (http://mafalda.fdi.ucm.es/madrid2015/), a week on
quantitative and formal methods for computer safety, reliability, and
performance.

31ST INTERNATIONAL CONFERENCE ON LOGIC
PROGRAMMING (ICLP 2015)
Call for Papers
Cork, Ireland
August 31 - September 4, 2015
http://booleconferences.ucc.ie/iclp2015

* HISTORY
Since the first conference held in Marseilles in 1982, ICLP has been the premier international conference for presenting research in logic programming. ICLP 2015 will be co-located with the 21st International Conference on Principles and Practice of Constraint Programming (CP 2015) and is part of "The Year of George Boole", a celebration of the life and work of George Boole who was born in 1815 and worked at the University College of Cork.

* IMPORTANT DATES
Abstracts due: April 20, 2015
Papers due: April 27, 2015
Notification to authors: June 5, 2015
Camera ready versions due: July 21, 2015
Conference: August 31-September 4, 2015

* CONFERENCE SCOPE
Contributions are sought in all areas of logic programming, including but not restricted to:
- Theory: Semantic Foundations, Formalisms, Nonmonotonic Reasoning, Knowledge Representation.
- Implementation: Compilation, Virtual Machines, Parallelism, Constraint Handling Rules and Tabling.
- Environments: Program Analysis, Transformation, Validation, Verification, Debugging, Profiling, Testing.
- Language Issues: Concurrency, Objects, Coordination, Mobility, Higher Order, Types, Modes, Assertions, Programming Techniques.
- Related Paradigms: Inductive and Coinductive Logic Programming, Constraint Logic Programming, Answer-Set Programming, SAT, Constraints, Computational Argumentation, Abductive Logic Programming, Functional Logic Programming.
- Applications: Databases, Data Integration and Federation, Software Engineering, Natural Language Processing, Web and Semantic Web, Agents, Artificial Intelligence, Bioinformatics, Social Networks and Social Choice.

In addition to the presentations of accepted papers, the technical program will include invited talks, advanced tutorials, the doctoral consortium, the Prolog contest and several workshops.

* SUBMISSION DETAILS
There are two categories for submissions:

-- Regular papers, including: (1) technical papers for describing technically sound, innovative ideas that can advance the state of logic programming; (2) application papers, with emphasis on impact on some application domains; (3) system and tool papers, with emphasis on novelty, practicality, usability and availability of the systems and tools described.

-- Technical communications aimed at describing recent developments, new projects, and other materials not ready for publication as regular papers.

All regular papers and technical communications will be presented during the conference. All submissions must be written in English and describe original, previously unpublished research, and must not simultaneously be submitted for publication elsewhere.

Regular papers must not exceed 12 pages plus bibliography: however the papers may include appendices beyond 12 pages. Technical communications must not exceed 10 pages. Submissions must be made in the TPLP format (see http://journals.cambridge.org/images/fileUpload/images/tlp˙ifc˙MAY2014.pdf) via the EasyChair submission system, available at www.easychair.org/conferences/?conf=iclp2015.

* PAPER PUBLICATION

All accepted regular papers will be published in the journal Theory and Practice of Logic Programming (TPLP), Cambridge University Press (CUP), in one or more special issues. In order to ensure the quality of the final version, papers may be subject to two rounds of refereeing (within the decision period). Accepted technical communications will be published in archival form. The program committee may also recommend papers submitted as regular to be published as technical communications.

* ICLP 2015 ORGANIZATION

- General Co-Chairs:
  Barry O'Sullivan University College Cork, Ireland
  Roland Yap National University of Singapore
- Program Co-Chairs:
  Thomas Eiter TU Wien, Austria
  Francesca Toni Imperial College London, UK
- Local Arrangements Co-Chairs:
  Barry O'Sullivan University College Cork, Ireland
  Ken Brown University College Cork, Ireland
- Workshops Chair:
  Mats Carlsson SICS, Uppsala, Sweden
- Doctoral Consortium Chairs:
  Marina De Vos University of Bath, UK
  Yuliya Lierler University of Nebraska at Omaha, USA
- LP/CP Programming Contest Chair:
  Neng-Fa Zhou, City University of New York, USA
  Peter Stuckey, NICTA and the University of Melbourne, Australia
- Publicity Chair:
  Ian Miguel University of St Andrews, UK

13TH INTERNATIONAL SYMPOSIUM ON AUTOMATED TECHNOLOGY FOR

VERIFICATION AND ANALYSIS (ATVA 2015)
October 12-15, 2015
Shanghai, China
http://atva2015.ios.ac.cn/

* BACKGROUND

The purpose of ATVA is to promote research on theoretical and practical aspects of automated analysis, verification and syn-thesis by providing a forum for interaction between the regional and the international research communities and industry in the field.

* SCOPE

ATVA 2015 solicits high-quality submissions in areas related to the theory and practice of automated analysis and verification of hardware and software systems. Topics of interest include, but are not limited to:

- Formalisms for modeling hardware, software and embedded systems
- Specification and verification of finite-state, infinite-state and parameterized systems
- Program analysis and software verification
- Analysis and verification of hardware circuits, systems-on-chip and embedded systems
- Analysis of real-time, hybrid, priced/weighted and probabilistic systems
- Deductive, algorithmic, compositional, and abstraction refinement techniques for analysis and verification
- Analytical techniques for safety, security, and dependability
- Testing and runtime analysis based on verification technology
- Analysis and verification of parallel and concurrent hardware/software systems
- Verification in industrial practice
- Applications and case studies

Theory papers should preferably be motivated by practical problems, and applications should be based on sound theory and should solve problems of practical interest.

* IMPORTANT DATES

April 22, 2015 Abstract submission deadline (AOE)
April 25, 2015 Paper submission deadline (AOE)
May 5, 2015 Submission of workshop proposals
Jun 8, 2015 Paper acceptance/rejection notification
Jun 10, 2015 Announcement of the accepted papers
July 5, 2015 Camera-ready copy deadline

* GENERAL CHAIR

Jifeng He (East China Normal University, China)

* PROGRAMME CHAIRS

Bernd Finkbeiner (Saarland University, Germany)
Geguang Pu (East China Normal University, China)
Lijun Zhang (Institute of Software, Chinese Academy of Sciences)

* PUBLICITY CHAIRS

David N. Jansen (Radboud Universiteit, Netherlands)
Huibiao Zhu (East China Normal University, China)

* WORKSHOP CHAIR

Jun Sun (National University of Singapore, SG)

* KEYNOTES

Dino Distefano (Queen Mary, University of London, UK)
Joost-Pieter Katoen (RWTH Aachen University, Germany)
Jay Strother Moore (University of Texas-Austin, USA)

20TH INTERNATIONAL SYMPOSIUM ON FUNDAMENTALS OF COMPUTATION THEORY (FCT 2015)

Call for Papers
August 17-19, 2015
Gdansk, Poland
http://sites.google.com/site/fct2015gdansk/

* SUBMISSION

Authors are invited to submit original research papers in all areas related to the Foundations of Computer Science (algorithms, formal methods, emerging fields of study). Submissions should be made via the conference website, no later than April 25, 2015 (anytime in the world). No simultaneous submission to other conferences with published proceedings is allowed.

* GENERAL

FCT is a biennial series of conferences in the field of theoretical computer science. It was established in 1977 for researchers interested in all aspects of theoretical computer science, and in particular algorithms, complexity, formal and logical methods.

The 20th edition of FCT will be held in Gdansk, a 1000-year old city on the Baltic coast in northern Poland, with a medieval Old Town picturesquely set on the banks of the Motlawa River.

* INVITED SPEAKERS

Marek Karpinski, Peter Widmayer, and Antonin Kucera.

* IMPORTANT DATES

Submission deadline: April 25, 2015
Notification to authors: June 4, 2015
Symposium: August 17-19, 2015

* SCOPE

The program committee is soliciting original and significant research contributions to the Fundamentals of Computation Theory, including (but not limited to) algorithms, formal methods and emerging fields (ad hoc, dynamic and evolving systems; algorithmic game theory; computational biology; foundations of cloud computing and ubiquitous systems; quantum information and quantum computing).

* PUBLICATION

Conference proceedings will be published in the Springer "Lecture Notes in Computer Science" series. Selected papers will be invited to a special issue of the journal "Discrete Mathematics & Theoretical Computer Science", devoted to FCT'15. It is required that each accepted paper be presented at the conference by one of its authors.

* PROGRAM CHAIRS

Adrian Kosowski, Inria and Universite Paris Diderot (co-chair)
Igor Walukiewicz, CNRS and Universite de Bordeaux (co-chair)

WEIGHTED LOGICS FOR AI: LOGIC, UNCERTAIN BELIEFS, PREFERENCES, PARTIAL TRUTH (WL4AI'15)

Call for Papers
July 25-27, 2015
Buenos Aires, Argentina
http://www.iiia.csic.es/wl4ai-2015/

* WEIGHTED LOGICS

In this workshop, continuation of the successful workshops with the same name held at ECAI-2012 and IJCAI-2013, the aim is to bring together researchers to discuss about the different motivations for the use of weighted logics in AI, the different types of calculi that are appropriate for these needs, and the problems that arise when putting them at work. Any paper on a weighted logical system in relation to any of the following topics (but not limited to) with an AI perspective is welcome:

- weighted argumentation systems
- uncertain extensions of description logics
- logical aspects of graded BDI agents
- graded emotions
- graded truth
- belief revision in weighted logics
- inconsistency handling in weighted logics
- information fusion in weighted logics
- proof systems and decision procedures for weighted logics
- uncertainty extensions of logic programs
- weighted systems and non monotonic reasoning
- preference modeling
- logics of graded trust and reputation

Authors are especially encouraged to discuss the intended semantics of the weights they use in their paper. Details about submission guidelines will be posted shortly.

* IMPORTANT DATES (TENTATIVE)

- April 27, 2015: Deadline for submission of contributions to the workshop
- May 20, 2015: Notification on workshop paper submissions
- May 30, 2015: Deadline for sending the final camera ready copy to workshop organizers

* WORKSHOP CO-CHAIRS

Marcelo Finger, University of Sao Paulo, Brazil
Lluis Godo, IIIA-CSIC, Spain
Henri Prade, IRIT-CNRS, France
Guilin Qi, Southeast University, China

7TH NASA FORMAL METHODS SYMPOSIUM

Call for Participation
April 27-29, 2015
http://www.NASAFormalMethods.org/nfm2015
Pasadena, California, USA

* THEME

The widespread use and increasing complexity of mission- and safety-critical systems require advanced techniques that address their specification, verification, validation, and certification. The NASA Formal Methods Symposium is a forum for theoreticians and practitioners from academia, industry, and government, with the goals of identifying challenges and providing solutions to achieving assurance in mission- and safety-critical systems. The focus of the symposium is on formal methods, and aims to foster collaboration between NASA researchers and engineers and the wider aerospace and academic formal methods communities.

* TOPICS
  Topics of interest include, but are not limited to:
  Model checking, Theorem proving, SAT and SMT solving, Symbolic execution, Static analysis, Runtime verification, Systematic testing, Program refinement, Compositional verification, Modeling and specification formalisms, Model-based development, Model-based testing, Requirement engineering, Formal approaches to fault tolerance, Security and intrusion detection, Applications of formal methods
* INVITED SPEAKERS
  - Dino Distefano
  Software Engineer at Facebook, California, USA
  Professor at Queen Mary University of London, UK.
  - Viktor Kuncak
  Leads Lab for Automated Reasoning and Analysis at EPFL, Lausanne, Switzerland.
  - Rob Manning
  Chief Engineer at NASA/JPL.
* LOCATION, COST, REGISTRATION AND HOTEL ROOM BOOKING
  The symposium will take place at the Hilton Hotel, Pasadena, California, USA
  There will be no registration fee for participants.
  All interested individuals, including non-US citizens, are welcome to attend; however, all attendees must register (but please only register if you intend to attend). Registration form and hotel booking websites are reachable from the main website. A block of rooms at a low price are reserved with booking deadline of March 26.
* PC CHAIRS
  Klaus Havelund, NASA Jet Propulsion Laboratory, USA
  Gerard Holzmann, NASA Jet Propulsion Laboratory, USA
  Rajeev Joshi, NASA Jet Propulsion Laboratory, USA

E. W. BETH DISSERTATION PRIZE 2015
  Call for Nominations
  http://www.folli.info/?page˙id=84
* HISTORY
  Since 2002, FoLLI (the Association for Logic, Language, and Information) has awarded the E.W. Beth Dissertation Prize to outstanding dissertations in the fields of Logic, Language, and Information.
  We invite submissions for the best dissertation which resulted in a Ph.D. degree awarded in 2014. The prize will be awarded at the ESSLLI summer school in Barcelona.
* WHO QUALIFIES
  Nominations of candidates are admitted who were awarded a Ph.D. degree in the areas of Logic, Language, or Information between January 1st, 2014 and December 31st, 2014. Theses must be written in English; however, the Committee accepts submissions of English translations of theses originally written in other languages, and for which a PhD was awarded in the preceding two years (i.e. between January 1st, 2012 and December 31st, 2013). There is no restriction on the nationality of the candidate or on the university where the Ph.D. was granted.
* PRIZE
  The prize consists of:

-a certificate
-a donation of 2500 euros provided by the E.W. Beth Foundation
-an invitation to submit the thesis (or a revised version of it) to the FoLLI Publications on Logic, Language and Information (Springer).

* HOW TO SUBMIT
See http://www.folli.info/?page_id=84
Queries: Ian Pratt-Hartmann (ipratt@cs.man.ac.uk)
* IMPORTANT DATES
Deadline for Submissions: April 27th, 2015.
Notification of Decision: July 6th, 2015.
ESSLLI summer school: August 3rd-14th, 2015
* COMMITTEE
Raffaella Bernardi (Trento), Johan Bos (Groningen), Julian Bradfield (Edinburgh), Wojciech Buszkowski (Poznan), Michael Kaminski (Technion, Haifa), Marco Kuhlmann (Linkoping), Larry Moss (Bloomington), Valeria de Paiva (Nuance Communications), Ian Pratt-Hartmann (chair) (Manchester), Ruy de Queiroz (Recife), Mehrnoosh Sadrzadeh (Queen Mary, London), Rineke Verbrugge (Groningen)

11TH INTERNATIONAL WORKSHOP ON LOGICAL FRAMEWORKS AND META-LANGUAGES: THEORY AND PRACTICE (LFMTP 2015)
Call for papers
1 August 2015
Berlin, Germany
Affiliated with CADE-25
http://lfmtp.org/2015
* LFMTP is an annual forum where researchers in the area of type theories, logical frameworks and meta-languages report on recent foundational and applied advances in these fields. The workshop brings together designers, implementers, and practitioners working on these areas.
* IMPORTANT DATES
- 30 April 2015: abstract submission
- 7 May 2015: paper submission
- 12 June 2015: notifications to authors
* Detailed information can be found on the webpage.

22ND INTERNATIONAL SYMPOSIUM ON TEMPORAL REPRESENTATION AND REASONING (TIME 2015)
Call for papers
Symposium on September 23 - 25, 2015, Kassel Germany
http://time2015.uni-kassel.de
Submission deadline on April 30th
* TIME 2015 aims to bring together researchers interested in reasoning about temporal aspects of information in any area of Computer Science. The symposium has a wide remit and intends to cater both for theoretical aspects and well-founded applications. One of the key aspects of the time symposium is its interdisciplinarity with attendees from distinct areas such as artificial intelligence, database management, logic and verification, and beyond. The symposium will encompass three tracks on temporal representation and reasoning in AI, Databases, as well as Logic and Verification.

* Submissions of high quality papers describing research results are solicited. Submitted papers should contain original, previously unpublished content, should be written in English, and must not be simultaneously submitted for publication elsewhere. Submitted papers will be refereed for quality, correctness, originality, and relevance.
* Invited Speakers: Giuseppe de Giacomo, Sapienza Universita di Roma; Carlo Zaniolo, UCLA; Benedikt Bollig, ENS Cachan
* Proceedings and Special Issue: Accepted papers will be presented at the symposium and included in the published by the IEEE Conference Publishing Services (CPS). The authors of the best papers of the conference will be invited to submit an extended version of their contribution to a special issue appearing in the journal Information & Computation.

DOMAINS XII

Call for Abstract and Participation
http://booleconferences.ucc.ie/gbmsc2015/domainsxii
25-28 Aug 2015

* DOMAINS
  The international conference Domains is aimed at computer scientists and mathematicians alike who share an interest in the mathematical foundations of computation. Previous Domains meetings include Paris (2014), Swansea (2011), Sussex (2008) and Novosibirsk (2007). Domains is an open and interactive event, welcoming talks on topics related to domains and related aspects (list included below) as well as talks pertaining to the general Boole theme of the overarching mathematical sciences conference. Conference publication will be considered pending participant interest. Domains XII will be embedded in the George Boole Mathematical Sciences (GMBS) Conference.
* INVITED SPEAKERS
  - Stephen Brookes (Carnegie Mellon)
  - Thierry Coquand (Gothenburg)
  - Abbas Edalat (Imperial College)
  - Jean Goubault-Larrecq (LSV, CNRS & ENS de Cachan)
  - Chris Heunen (Oxford)
  - Gordon Plotkin (Edinburgh)
  - Dana Scott (Carnegie Mellon)
* TOPICS
  Topics of interaction with domain theory for this workshop include, but are not limited to: program semantics, program logics, probabilistic computation, exact computation over the real numbers, lambda calculus, games, models of sequential computation, constructive mathematics, recursion theory, realizability, real analysis and computability, topology, metric spaces and domains, idempotent analysis and domains, locale theory, category theory, topos theory, type theory
* IMPORTANT DATES
  Registration: opened March 28
  Submission of Abstracts: by May 1
  Please send abstracts to: m.schellekens@cs.ucc.ie
  Notification of acceptance: June 1
  Late registration: as of June 15 to August 15

12TH INTERNATIONAL WORKSHOP ON QUANTUM PHYSICS AND LOGIC (QPL 2015)

Call for Papers and Participation
July 13-17, 2015
Oxford, United Kingdom
http://www.cs.ox.ac.uk/qpl2015

* SERIES
QPL is a workshop that brings together researchers working on mathematical foundations of quantum physics, quantum computing, spatio-temporal causal structures, and related areas such as computational linguistics. Of particular interest are topics that use logical tools, ordered algebraic and category-theoretic structures, formal languages, semantical methods and other computer science methods for the study of physical behaviour in general.
* IMPORTANT DATES
Submission deadline: 1 May 2015;
Notification: 1 June 2015
* INVITED SPEAKERS
Paul Busch (York), Dan Browne (London), Chris Douglas (Oxford)
* TUTORIALS
Paul Busch (York), Dan Browne (London), Oscar Dahlsten (Oxford), Pawel Sobocinski (Southampton)
* Registration is now open
* Detailed information can be found on the webpage.

10TH INTERNATIONAL SYMPOSIUM ON FRONTIERS OF COMBINING SYSTEMS (FroCoS 2015)

Call for Papers
Wroclaw, Poland
September 21-24, 2015
http://frocos2015.ii.uni.wroc.pl/

* GENERAL INFORMATION
The 10th International Symposium on Frontiers of Combining Systems (FroCoS 2015) will be held in Wroclow, Poland, from September 21 to September 24, 2015. Its main goal is to disseminate and promote progress in research areas related to the development of techniques for the integration, combination, and modularization of formal systems together with their analysis.
FroCos 2015 will be co-located with the 23rd International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2015) to be held also on September 21-24, 2015. The local organization of both events is handled by Hans de Nivelle.
* SCOPE OF CONFERENCE
FroCoS 2015 seeks to offer a common forum for research in the general area of combination, modularization, and integration of systems, with emphasis on logic-based ones, and of their practical use.
Typical topics of interest include (but are not limited to):
- combinations of logics (such as higher-order, first-order, temporal, modal, description or other non-classical logics);
- combination and integration methods in SAT and SMT solving;
- combination of decision procedures, satisfiability procedures, constraint

solving techniques, or logical frameworks;
- combinations and modularity in ontologies;
- integration of equational and other theories into deductive systems;
- hybrid methods for deduction, resolution and constraint propagation;
- hybrid systems in knowledge representation and natural language semantics;
- combined logics for distributed and multi-agent systems;
- logical aspects of combining and modularizing programs and specifications;
- integration of data structures into constraint logic programming and deduction;
- combinations and modularity in term rewriting;
- applications of methods and techniques to the verification and analysis of information systems.

* PUBLICATION DETAILS

The proceedings of the symposium will be published in the Springer LNAI/LNCS series.

* PAPER SUBMISSIONS

The program committee seeks high-quality submissions describing original work, written in English, not overlapping with published or simultaneously submitted work to a journal or conference with archival proceedings. Selection criteria include accuracy and originality of ideas, clarity and significance of results, and quality of presentation. The page limit in Springer LNCS style is 16 pages.

* IMPORTANT DATES

Deadline (abstracts): 04.05.2015
Deadline (full papers): 10.05.2015
Author notification: 22.06.2015
Final version due: 20.07.2015
Workshops/Tutorials: 19-20.09.2015
Conference: 21-24.09.2015

* PROGRAM CHAIRS

Carsten Lutz, University of Bremen [co-chair]
Silvio Ranise, Fondazione Bruno Kessler [co-chair]

JEWELS OF AUTOMATA: FROM MATHEMATICS TO APPLICATIONS (AUTOMATHA 2015)

Call for Participation
Leipzig, May 6 - 9, 2015
http://www.automatha.uni-leipzig.de

* GENERAL

The conference AutoMathA 2015 will survey a wide picture of research in automata theory and related mathematical fields. It will consist of 26 invited lectures which will describe significant progress over the past years, and will be a meeting point for both young and senior researchers to learn and to discuss about automata theory, its connections with mathematics and its applications. For young researchers the conference will serve as an Advanced Spring School of Automata Theory. All lectures will have the same length of about 50 minutes and will deal with significant progress over the past years. They will serve as surveys on such recent work, not necessarily focusing only on work of the speaker. There will be only informal proceedings (with slides or texts); so the results presented are usually already published (or accepted) elsewhere. This event is a follow-up of the multidisciplinary programme Automata: from Mathematics to Applications

(AutoMathA, 2005 - 2010) of the European Science Foundation and of its final conference ”Highlights of AutoMathA” in Vienna 2010.

* INVITED SPEAKERS

Jean-Paul Allouche (Paris), Mikolaj Bojanczyk (Warsaw), Patricia Bouyer-Decitre (Cachan), Thomas Brihaye (Mons), Olivier Carton (Paris), Thomas Colcombet (Paris), Volker Diekert (Stuttgart), Michael Elberfeld (Aachen), Uli Fahrenberg (Rennes), Nathanael Fijalkow (Paris), Paul Gastin (Cachan), Artur Jez (Saarbruecken), Ines Klimann (Paris), Dexter Kozen (Ithaca), Dietrich Kuske (Ilmenau), Kim Larsen (Aalborg), Ranko Lazic (Warwick), Markus Lohrey (Siegen), Cyril Nicaud (Marne-la-Vallee), Igor Potapov (Liverpool), Jacques Sakarovitch (Paris), Thomas Schwentick (Dortmund), Marinella Sciortino (Palermo), Ludwig Staiger (Halle), Moshe Vardi (Houston), James Worrell (Oxford)

* POSTER

http://www.automatha.uni-leipzig.de/AutoMathA.pdf

QUANTIFY 2015 - 2nd INTERNATIONAL WORKSHOP ON QUANTIFICATION

Call for Papers
August 1 2015
Berlin, Germany
http://fmv.jku.at/quantify15/
Affiliated to and co-located with CADE 2015

* AIM

The goal of QUANTIFY 2015 is to provide an interdisciplinary forum for researchers who investigate the impact of quantification from a theoretical as well as from a practical point of view. The workshop is concerned with all aspects of quantification in logics such as QBF, QCSP, SMT, and theorem proving.

* TOPICS

Topics of Interest: complexity results, encodings with and without quantification and comparisons thereof, applications of quantification, implementations of reasoning tools, case studies and experimental results, intersections between the different research communities working on quantification, surveys of state of the art approaches to handling quantification

* IMPORTANT DATES

May 8 2015: paper submission
May 29 2015: notification of acceptance
June 23 2015: camera-ready version of papers
August 1 2015: workshop

* INVITED SPEAKER

Olaf Beyersdorff, University of Leeds

* SUBMISSION

Please see the webpage http://fmv.jku.at/quantify15/ for further information.

FIFTH SUMMER SCHOOL ON FORMAL TECHNIQUES (SSFT'15)

May 17-22, 2015
Menlo College, Atherton, CA, USA
http://fm.csl.sri.com/SSFT15

* GENERAL
  Techniques based on formal logic, such as model checking, satisfiability, static analysis, and automated theorem proving, are finding a broad range of applications in modeling, analysis, verification, and synthesis. This school, the fifth in the series, will focus on the principles and practice of formal techniques, with a strong emphasis on the hands-on use and development of this technology. It primarily targets graduate students and young researchers who are interested in studying and using formal techniques in their research. A prior background in formal methods is helpful but not required. Participants at the school will have a seriously fun time experimenting with the tools and techniques presented in the lectures during laboratory sessions.
* LECTURERS
  The lecturers at the school include: Arie Gurfinkel (SEI CMU), Cathy Meadows (NRL), Bart Jacobs (KU Leuven), and Kim Guldstrand Larsen (Aalborg University, Denmark). We expect to support travel, food, and accomodation for students registered at US universities. Others will be charged around US$550. All are encouraged to apply at http://fm.csl.sri.com/SSFT15

12TH INTERNATIONAL COLLOQUIUM ON THEORETICAL ASPECTS OF COMPUTING (ICTAC 2015)
  Call for Papers
  29-31 October 2015
  Cali, Colombia
  http://www.ictac2015.co
* SERIES
  The ICTAC conference series aims at bringing together practitioners and researchers to exchange ideas and experiences addressing challenges in theoretical aspects of computing as well as in exploiting theory through methods and tools for system development. ICTAC also aims to promote cooperation between participants and institutions from developing and industrial countries in research and education.
* THEMES AND TOPICS OF PAPERS
  Topics of interest include theories of computation and programming, foundations of software engineering and formal techniques in software design and verification, as well as tools that support formal techniques for software modeling, system design and verification.
* INVITED SPEAKERS
  Jean-Raymond Abrial
  Volker Diekert
  Cesar Munoz
  Catuscia Palamidessi
  Davide Sangiorgi
  Moshe Vardi
  Glynn Winskel
* ASSOCIATED EVENTS
  - ICTAC Summer School on Formal Methods (October 25-27)
  - DCM 2015: 11th International Workshop on Developments in Computational Models (October 28)
* PAPER SUBMISSION

Important Dates
- Abstract submission: Monday, June 1, 2015.
- Paper submission: Friday, June 5, 2015.
- Author notification: Monday, July 20, 2015.
- Camera ready: Monday, August 3, 2015.

* PAPER CATEGORIES AND FORMAT

We call for submissions, related to the above areas and topics, according to the following three categories:
- Regular papers, with original research contributions;
- Short papers, on recent work or proposals of emerging challenges;
- Tool papers, on tools that support formal techniques for software modeling, system design and verification. Submissions should adhere to the LNCS format (see http://www.springer.de/comp/lncs/authors.html for details). Regular papers should not exceed 18 pages. Short and tool papers should not exceed 10 pages.

Submissions to the colloquium must not have been published or be concurrently considered for publication elsewhere. All submissions will be judged on the basis of originality, contribution to the field, technical and presentation quality, as well as their relevance to the conference.
Papers must be submitted by using www.easychair.org/conferences/?conf=ictac2015.

* PROCEEDINGS

As for the past editions, the proceedings of ICTAC 2015 will be published by Springer in the series Lecture Notes in Computer Science (LNCS).

* SPECIAL ISSUE

Extended versions of selected papers from ICTAC 2015 will be invited to a special issue of Mathematical Structures in Computer Science (http://journals.cambridge.org/action/displayJournal?jid=MSC).

* CHAIRS
- Martin Leucker, University of Luebeck (DE).
- Camilo Rueda, Pontificia Universidad Javeriana - Cali (CO).
- Frank Valencia, CNRS and LIX, Ecole Polytechnique (FR).

* CONTACT CHAIR
- Frank D. Valencia (frank.valencia@gmail.com).

SYMPOSIUM ON DEPENDABLE SOFTWARE ENGINEERING: THEORIES, TOOLS AND APPLICATIONS (SETTA 2015)

Call for Papers
November 4-6, 2015
Nanjing University
http://cs.nju.edu.cn/setta/

* BACKGROUND AND OBJECTIVES

The aim of the symposium is to bring together international researchers and practitioners in the field of software technology. Its focus is on formal methods and advanced software technologies, especially for engineering complex, large-scale artefacts like cyber-physical systems, networks of things, enterprise systems, or cloud-based services. Contributions relating to formal methods or integrating them with software engineering, as well as papers advancing scalability or widening the scope of rigorous methods to new design goals are especially welcome. Being hosted in China, the symposium will also provide a platform for building up research collaborations between the rapidly growing Chinese computer science community and its international counterpart. The symposium will support this process through dedicated events and therefore welcomes both young researchers considering international collaboration in formal methods and established researchers looking for international cooperation and willing to attract new colleagues to the domain.

* SUBMISSIONS

Authors are invited to submit papers on original research, industrial applications, or position papers proposing challenges in fundamental research and technology. The latter two types of submissions are expected to contribute to the development of formal methods either by substantiating the advantages of integrating formal methods into the development cycle or through delineating need for research by demonstrating weaknesses of existing technologies, especially when addressing new application domains.
Submissions can take the form of either normal or short papers. Short papers can discuss ongoing research at an early stage, including PhD projects.
Papers should be written in English. Regular Papers should not exceed 15 pages and Short Papers should not exceed 6 pages in LNCS format (see http://www.springer.de/comp/lncs/authors.html for details).
The proceedings will be published as a volume in Springer's LNCS series.
The authors of a selected subset of accepted papers will be invited to submit extended versions of their papers to appear in a special issue of the Formal Aspect Computing journal.

* TOPICS

- Requirements specification and analysis
- Formalisms for modeling, design and implementation
- Model checking, theorem proving, and decision procedures
- Scalable approaches to formal system analysis
- Formal approaches to simulation and testing
- Integration of formal methods into software engineering practice
- Contract-based engineering of components, systems, and systems of systems
- Formal and engineering aspects of software evolution and maintenance
- Parallel and multicore programming
- Embedded, real-time, hybrid, and cyber-physical systems
- Mixed-critical applications and systems
- Formal aspects of service-oriented and cloud computing
- Safety, reliability, robustness, and fault-tolerance
- Empirical analysis techniques and integration with formal methods
- Applications and industrial experience reports
- Tool integration

* IMPORTANT DATES

June 12,2015 Abstracts
June 19,2015 Submission of papers
August 21,2015 Notification to authors
September 4,2015 Camera-ready versions

* KEYNOTE SPEAKERS

Sanjoy Baruah, University of North Carolina at Chapel Hill, USA
David Harel,   Weizmann Institute of Science, Israel
Huimin Lin,   Institute of Software, CAS, China

* GENERAL CHAIR

Jian Lv, Nanjing University, China

* PROGRAMME CO-CHAIRS

Xuandong Li, Nanjing University, China

Zhiming Liu, Birmingham City University, UK
Yi Wang, Uppsala University, Sweden
* PUBLICITY CHAIRS
Jonathan Bowen, Birmingham City University, UK
Lijun Zhang, Institute of Software,Chinese Academy of Sciences, China
* PUBLICATION CHAIR
Martin Fraenzle, University of Oldenburg, Germany
* LOCAL ORGANIZATION CHAIR
Xin Chen, Nanjing University, China

THIRD INTERNATIONAL CONFERENCE FOR THE HISTORY AND PHILOSOPHY (HaPoC 3)
Call For Papers
8-11 October, 2015, Pisa
http://hapoc2015.di.unipi.it
* GENERAL
The DHST commission for the history and philosophy of computing (www.hapoc.org) is happy to announce the third HAPOC conference. The series aims at creating an interdisciplinary focus on computing, stimulating a dialogue between the historical and philosophical viewpoints.
* TOPICS
Topics include but are not limited to
- History and Philosophy of Computation (interpretation of the Church-Turing thesis; models of computation; logical/mathematical foundations of computer science; information theory...)
- History and Philosophy of Programming (classes of programming languages; philosophical status of programming...)
- History and Philosophy of the Computer (from calculating machines to the future of the computer; user interfaces; abstract architectures...)
- History and Epistemology of the use of Computing in the sciences (simulation vs. modelisation; computer-assisted proofs; linguistics...)
- Computing and the Arts: historical and conceptual issues (temporality in digital art; narration in interactive art work...)
- Social, ethical and pedagogical aspects of Computing (pedagogy of computer science; algorithms and copyright; Internet, culture, society...)
* INVITED SPEAKERS
Nicola Angius (Universita di Sassari, IT),
Lenore Blum (Carnagie Mellon University, USA),
David Allan Grier (IEEE & George Washington University, USA),
Furio Honsell (Universita di Udine, IT),
Pierre Mounier-Kuhn (CNRS & Universite Paris-Sorbonne, F),
Franck Varenne (Universit de Rouen, F).
* SUBMISSIONS
We cordially invite researchers working in a field relevant to the topics of the conference to submit a short abstract of approximately 200 words and an extended abstract of at most a 1000 words (references included) to www.easychair.org/conferences/?conf=hapoc2015
Abstracts must be written in English and anonymised.
Please check out the website of HaPoC 2015 for more information on the conference at http://hapoc2015.sciencesconf.org

A post-proceedings volume is going to appear in the IFIP Advances in Information and Communication Technology series, published by Springer.

* IMPORTANT DATES
  Submission deadline: June 19, 2015
  Notification of acceptance: July 19, 2015

22ND WORKSHOP ON LOGIC, LANGUAGE, INFORMATION AND COMPUTATION (WoLLIC 2015)

Call for Participation
July 20th-23rd, 2015, Bloomington, IN, USA
http://www.indiana.edu/~iulg/wollic

* AIMS
  WoLLIC is an annual international forum on inter-disciplinary research involving formal logic, computing and programming theory, and natural language and reasoning. Each meeting includes invited talks and tutorials as well as contributed papers.
* INVITED SPEAKERS
  Adriana Compagnoni (Stevens Institute, USA)
  Nina Gierasimczuk (University of Amsterdam)
  John Harrison (Intel, USA)
  Peter Jipsen (Chapman U, USA)
  Andre Joyal (U du Qubec Montreal, Canada)
  Chung-chieh Shan (Indiana U, USA)
  Alexandra Silva (Radboud U Nijmegen, The Netherlands)
  Mehrnoosh Sadrzadeh (Queen Mary, UK)
* STUDENT GRANTS
  ASL sponsorship of WoLLIC 2015 will permit ASL student members to apply for a modest travel grant (deadline: May 1st, 2015).
* DETAIS
  Detailed information can be found on the webpage
  www.indiana.edu/~iulg/wollic

GEORGE BOOLE MATHEMATICAL SCIENCES CONFERENCE (GBMSC 2015)

Call for Participation
17-25 August 2015
Cork, Ireland
http://booleconferences.ucc.ie/gbmsc2015

* CONTEXT
  As part of the celebrations of Boole's bicentenary, the George Boole Mathematical Sciences (GBMS) Conference (including Domains XII) will be held in University College Cork (UCC) during the last two weeks of August 2015. George Boole (1815-1864) was the first professor of mathematics at Cork. Boole's efforts to mathematize logical thinking caused a lasting paradigm shift in the 19th century which enlarged the scope and potency of modern mathematics, and provided a wealth of ideas for applications in diverse scientific areas resulting in ground-breaking innovations during the 20th century and beyond. This event will include 100-150 lectures on selected areas:
  - Theme 1: Boole and Beyond in Quantum Information Theory
  - Theme 2: From Boole's Algebra of Logic to Boolean Algebra, and Beyond

- Theme 3: Complex and Boolean NetworksGeometry and Visualization
- Theme 4: Geometry and Visualization
- Theme 5: Harmonic Analysis
- Theme 6: Invariants from Moduli Spaces
- Theme 7: Mathematical Financial Modelling Post-Crisis
- Theme 8: Quantum Probabilistic Symmetries and Quantized Boolean Algebras

The following events are embedded in and associated with the conference:
- 2015 Annual Meeting of the Irish Mathematical Society (IMS)
- Domains XII
- When Boole Meets Shannon

GBMS THEME 2: FROM BOOLE'S ALGEBRA OF LOGIC TO BOOLEAN ALGEBRA, AND BEYOND

Call for Participation
27-28 August 2015
Cork, Ireland
http://booleconferences.ucc.ie/gbmsc2015/theme2

* CONTEXT

The international conference "From Boole's Algebra of Logic to Boolean Algebra, and Beyond" will take place during the celebrations of Boole's bicentenary at University Cork. As an open and interactive event, the conference welcomes talks on topics related to the main theme, Boole's Algebra of Logic, Boolean Algebra as well as other developments emanating from Boole's work or presentations pertaining to the history of Boole's work and related discoveries. Conference publication will be considered pending participant interest.

"From Boole's Algebra of Logic to Boolean Algebra, and Beyond" will be embedded in the George Boole Mathematical Sciences (GMBS) Conference.

* INVITED SPEAKERS
- Stanley Burris (University of Waterloo)
- Amirouche Moktefi (Tallinn University of Technology)
- Don Monk (University of Colorado Boulder)
- Hillary Priestley (University of Oxford)
- Stevo Todorcevic (University of Toronto and CNRS, Paris)

* STEERING COMMITTEE
- Steven Givant (Mills College)
- Wilfrid Hodges (University of London)
- Angus Mcintyre (Queen Mary London)
- Michel Schellekens (UCC)
- Dana Scott (Carnegie Mellon)

* ORGANIZERS
- Aoife Hennessy (WIT)
- Michel Schellekens (UCC)

5TH INTERNATIONAL ABZ 2014 CONFERENCE (ASM, Alloy, B, TLA, VDM, Z)

Call for Papers, Answers to the case study, Workshops, Tutorials
May 23-27, 2016
Linz, Austria
http://www.cdcc.faw.jku.at/ABZ2016/

* The ABZ conference is dedicated to the cross-fertilization of six related

state-based and machine-based formal methods, Abstract State Machines (ASM), Alloy, B, TLA, VDM and Z. Contributions are solicited on all aspects of the theory and applications of ASMs, Alloy, B, TLA, VDM, Z approaches in software/hardware engineering, including the development of tools and industrial applications.

* Types of submission:
  -- Research papers: full papers of not more than 14 pages in LNCS format, which have to be original, unpublished and not submitted elsewhere.
  -- Short presentations of work in progress, and tool demonstrations. An extended abstract of not more than 4 pages is expected and will be reviewed.
  -- Answers to case study papers: full papers of not more than 14 pages in LNCS format reporting on the experiments conducted with any of the state based techniques in the scope of ABZ 2014.
  -- Application in industry papers reporting on work or experiences on the application of state based formal methods in industry. An extended abstract of not more than 4 pages is expected and will be reviewed.
* Submission site: https://easychair.org/conferences/?conf=abz2016
* Important Dates:
  Workshop proposal submission: October 16, 2015
  Research paper, Answers to case study submission: January 15, 2016
  Short and industry paper submission: February 4, 2016
  Tutorial proposal submissions: February 15, 2016
  Tutorial proposal notifications: March 14, 2016
* Detailed information can be found on the conference website
* Contact: Klaus-Dieter SCHEWE (klaus-dieter.schewe@scch.at)

PHD STUDENTSHIP IN ALGORITHMS AND COMPLEXITY AT ROYAL HOLLOWAY, UNIVERSITY OF LONDON
  http://www.cs.rhul.ac.uk/home/tzameret/PhD˙Post.html
* The Department of Computer Science at Royal Holloway, University of London is offering a three-year full-time PhD studentship in algorithms and complexity starting in October 2015. The studentship includes a full tuition-fee waiver and a maintenance award in line with the level recommended by Research Councils UK (exact value to be confirmed, circa GBP 16,000 (USD 25,000 or EUR 20,000)). The student will be hosted in the Center for Algorithms and Applications and will work under the supervision of Dr Iddo Tzameret (http://www.cs.rhul.ac.uk/home/tzameret/).
* THE PROJECT is broadly in the area of computational complexity with an emphasis on satisfiability and the complexity of proofs. The successful candidate will investigate fundamental aspects of the Boolean satisfiability problem SAT from possibly different aspects - combinatorial, algebraic and logical - with a possibility to engage as well in applied or empirical study of SAT-solving and other applications related to SAT, depending on the preferences and qualifications of the candidate.
* For more information about the post see:
  http://www.cs.rhul.ac.uk/home/tzameret/PhD˙Post.html
* STARTING DATE: October 1, 2015
* HOW TO APPLY:

Applications should be made as soon as possible through the
online application system at Royal Holloway, University of London:
https://www.royalholloway.ac.uk/studyhere/researchdegrees/applying/home.aspx
and will remain open until the position is filled.
* For any informal inquiries about the position, please contact
Dr Iddo Tzameret at: Iddo.Tzameret@rhul.ac.uk

NEW DOCTORAL PROGRAM ON LOGICAL METHODS IN
COMPUTER SCIENCE (LogiCS)
http://logic-cs.at/phd
* Funded Doctoral Positions in Computer Science
* TU Wien, TU Graz, and JKU Linz are seeking exceptionally talented and
motivated students for their joint doctoral program LogiCS. The LogiCS
doctoral college focuses on interdisciplinary research topics covering
(i) computational logic, and applications of logic to
(ii) databases and artificial intelligence as well as to
(iii) computer-aided verification.
* THE PROGRAM
LogiCS is a doctoral college focusing on logic and its applications in
computer science. Successful applicants will work with and be
supervised by leading researchers in the fields of computational
logic, databases and knowledge representation, and computer-aided
verification.
* FACULTY MEMBERS

M. Baaz	A. Biere	R. Bloem	A. Ciabattoni
U. Egly	T. Eiter	C. Fermueller	R. Grosu
A. Leitsch	M. Ortiz	R. Pichler	S. Szeider
H. Tompits	H. Veith	G. Weissenbacher	

* POSITIONS AND FUNDING
We are looking for 1-2 doctoral students per faculty member, where 30%
of the positions are reserved for highly qualified female
candidates. The doctoral positions are funded for a period of 3 years
according to the funding scheme of the Austrian Science Fund
(details: http://www.fwf.ac.at/de/projects/personalkostensaetze.html)
The funding can be extended for one additional year contingent on a
placement at one of our international partner institutions.
* CURRENT RESEARCH AREAS
At the moment we are particularly looking for people in the following areas:
- Answer Set Programming
- Model Checking
- Proof Theory and Automated Deduction
- QBF-solving
- Static Analysis and Abstract Interpretation
* HOW TO APPLY
Detailed information about the application process is available on the
LogiCS web-page http://logic-cs.at/phd/
The applicants are expected to have completed an excellent diploma or
master's degree in computer science, mathematics, or a related
field. Candidates with comparable achievements will be considered on a
case-by-case basis. Applications by the candidates need to be
submitted electronically.

Applications can be submitted at any time. Next screening: June 15, 2015.
* HIGHEST QUALITY OF LIFE
The Austrian cities Vienna, Graz, and Linz, located close to the Alps and surrounded by beautiful nature, provide an exceptionally high quality of life, with a vibrant cultural scene, numerous cultural events, world-famous historical sites, a large international community, a varied cuisine and famous coffee houses.
* For further information please contact: info@logic-cs.at