TABLE OF CONTENTS

General Information
1 From the Editor Andrzej Murawski
2 Chair’s Letter Prakash Panangaden

Technical Columns
3 Complexity Neil Immerman
25 Verification Ranko Lazić

Regular Features
47 Conference Reports Jorge A. Pérez
54 SIGLOG Monthly 201 Daniela Petrișan
# SIGLOG NEWS

Published by the ACM Special Interest Group on Logic and Computation

## SIGLOG Executive Committee

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chair</td>
<td>Prakash Panangaden</td>
<td>McGill University</td>
</tr>
<tr>
<td>Vice-Chair</td>
<td>Luke Ong</td>
<td>University of Oxford</td>
</tr>
<tr>
<td>Treasurer</td>
<td>Amy Felty</td>
<td>University of Ottawa</td>
</tr>
<tr>
<td>Secretary</td>
<td>Alexandra Silva</td>
<td>University College London</td>
</tr>
<tr>
<td>EATCS President</td>
<td>Paul Spirakis</td>
<td>Universities of Liverpool and Patras</td>
</tr>
<tr>
<td>EACSL President</td>
<td>Thomas Schwentick</td>
<td>Technische Universität Dortmund</td>
</tr>
<tr>
<td>ACM ToCL E-in-C</td>
<td>Orna Kupferman</td>
<td>Hebrew University</td>
</tr>
<tr>
<td></td>
<td>Véronique Cortier</td>
<td>CNRS and LORIA, Nancy</td>
</tr>
<tr>
<td></td>
<td>Andrzej Murawski</td>
<td>University of Oxford</td>
</tr>
<tr>
<td></td>
<td>Catuscia Palamidessi</td>
<td>INRIA and LIX, École Polytechnique</td>
</tr>
</tbody>
</table>

## ADVISORY BOARD

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Martín Abadi</td>
<td>Google and UC Santa Cruz</td>
</tr>
<tr>
<td>Phokion Kolaitis</td>
<td>University of California, Santa Cruz</td>
</tr>
<tr>
<td>Dexter Kozen</td>
<td>Cornell University</td>
</tr>
<tr>
<td>Gordon Plotkin</td>
<td>University of Edinburgh</td>
</tr>
<tr>
<td>Moshe Vardi</td>
<td>Rice University</td>
</tr>
</tbody>
</table>

## TECHNICAL COLUMN EDITORS

<table>
<thead>
<tr>
<th>Topic</th>
<th>Name</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automata</td>
<td>Mikolaj Bojańczyk</td>
<td>University of Warsaw</td>
</tr>
<tr>
<td>Complexity</td>
<td>Neil Immerman</td>
<td>University of Massachusetts Amherst</td>
</tr>
<tr>
<td>Security and Privacy</td>
<td>Matteo Maffei</td>
<td>Technische Universität Wien</td>
</tr>
<tr>
<td>Semantics</td>
<td>Mike Mislove</td>
<td>Tulane University</td>
</tr>
<tr>
<td>Verification</td>
<td>Ranko Lazić</td>
<td>University of Warwick</td>
</tr>
</tbody>
</table>

## REGULAR FEATURES

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conference Reports</td>
<td>Jorge A. Pérez</td>
<td>University of Groningen and CWI</td>
</tr>
<tr>
<td>SIGLOG Monthly</td>
<td>Daniela Petrișan</td>
<td>Université Paris Diderot</td>
</tr>
</tbody>
</table>

Notice to Contributing Authors to SIG Newsletters

By submitting your article for distribution in this Special Interest Group publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

— to publish in print on condition of acceptance by the editor
— to digitize and post your article in the electronic version of this publication
— to include the article in the ACM Digital Library and in any Digital Library related services
— to allow users to make a personal copy of the article for noncommercial, educational or research purposes

However, as a contributing author, you retain copyright to your article and ACM will refer requests for republication directly to you.

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

In this issue

– SIGLOG Chair reflects on the 2018 Federated Logic Conference (FLoC) and the recent SIG Governing Board meeting.

– Andrei Bulatov surveys the CSP Dichotomy Conjecture and its solution in Neil Immerman’s Complexity Column.

– The Verification column, edited by Ranko Lazić, features an overview of recent research into synthesis problems in Markov decision processes by Christel Baier and Clemens Dubslaff.

– Two reports can be found in Jorge A. Pérez’s Conference Reports section.
  — Agata Ciabattoni, Björn Lellmann, and Kees van Berkel report on the workshop “Deontic Reasoning: from Ancient Texts to Artificial Intelligence”, which took place in June in Vienna.
  — Daniele Ahmed, Katherine Fletcher, and Julian Gutierrez write about FLoC 2018, which took place in Oxford in July.

– As usual, we wrap up with the latest issue of SIGLOG Monthly, prepared by Daniela Petrişan.

Enjoy!

Andrzej Murawski
University of Oxford
SIGLOG News Editor
dereditor@siglog.org
I wrote my last Chair’s message while I was in Oxford with FLoC about to begin. This FLoC proved to be incredibly successful. Kudos to Marta Kwiatkowska, Daniel Kroening and, of course, Moshe Vardi for pulling off yet another great meeting. There were approximately 2000 participants, nearly 7 times the membership of SIGLOG! The meeting kicked off with a plenary talk by Peter O’Hearn who is the exemplar of theory making an impact on practice. It was great to see two Turing Award winners (Dana Scott and Tony Hoare) in the audience. The keynote for the first week was another brilliant talk given by another Turing Award winner: Shafi Goldwasser. It was a special pleasure for me to see Moshe Vardi win the Church Prize for his work with Feder on the dichotomy conjecture.

One of the interesting events to which I was invited was a special session of the Women in Logic meeting where the underrepresentation of women in the Logic and Computation community was discussed at length. A day later this was followed up by a LICS invited talk by Ursula Martin on the same theme. I don’t need to say that the underrepresentation of and barriers faced by women are a vexing problem that cannot be ignored or dismissed. Unfortunately, some members of the community are still in denial.

Just before I wrote this letter I attended a SIG Governing Board meeting in Chicago where I got to meet the chairs of the other SIGs as well as senior members of ACM: the President (Cherri Pancake) and the CEO, Vicki Hanson. A number of issues came up which I would like to bring to the attention of the SIGLOG community. First, there is a new Code of Ethics available on the ACM web site which includes a section on harassment. Second, SIGPLAN has been exploring the possibility of including carbon offset fees in conference registrations. The idea is that we, the community that attends conferences, should do something to mitigate the effect of all the conference-related travel that we do. Their idea is to add between 30 and 40 dollars to the conference fees which would then be paid to a carbon offset vendor. These are organizations that plant trees and do other things like carbon capture to offset the effect of the carbon being generated by air travel. It is far from being ready to put into practice but it is worth discussing this now amongst ourselves.

Remember the upcoming SIGLOG elections. I do hope many of you are considering standing for one of the offices. Catuscia Palamidessi is the nominating officer so please contact her if you are interested in running.

Best wishes for the New Year

Prakash Panangaden
McGill University
ACM SIGLOG Chair
In 1975, Ladner proved that computational complexity is dense in the sense that if
$C_0$ is strictly contained in $C_1$, for any two complexity classes satisfying standard
closure properties, then there is a complexity class $C_{0.5}$ strictly between them. However,
naturally occurring algorithmic problems tend to be complete for one of a tiny stable of
important complexity classes. This is often observed and leads to the great practical
usefulness of the study of algorithms and complexity.

In 1978, Schaefer considered a natural family of satisfiability problems and showed
that they exhibit a striking dichotomy phenomenon: each of these problems is ei-
ther NP complete, or solvable in P, i.e., none of them is complete for one of Ladner’s
intermediate complexity classes.

In 1993, Feder and Vardi, identified the set of Constraint Satisfaction Problems
(CSP) as an important natural subset of NP, and they announced their CSP Dichotomy
Conjecture: every CSP over a finite domain satisfies Schaefer’s dichotomy; it is either
NP Complete or in P. Feder & Vardi’s conjecture was already known to be true for the
domain of size two, i.e., the boolean domain, by Schaefer’s Theorem.

Feder and Vardi’s conjecture has lead to 25 years of rich and deep work, on this and
related problems, culminating quite recently with the final pieces solved independently
by Bulatov and Zhuk.

I am grateful to Andrei Bulatov for the following beautiful survey of the CSP Di-
chotomy Conjecture and its solution.
In this paper we briefly survey the history of the Dichotomy Conjecture for the Constraint Satisfaction problem, that was posed 25 years ago by Feder and Vardi. We outline some of the approaches to this conjecture, and then describe an algorithm that yields an answer to the conjecture.

1. THE COMPLEXITY DICHOTOMY PHENOMENON

It is well known that if $P \neq NP$, then there are infinitely many distinct complexity classes between them [Ladner 1975]. However, the only known examples of such classes are constructed through diagonalization, and therefore are somewhat artificial. It is thus an appealing idea to suggest that no ‘natural’ problem attains any of those intermediate complexity classes and either belongs to P or is NP-complete. Proving such a result for all ‘natural’ problems may be difficult, as it is not clear what a ‘natural’ problem may mean precisely. However, in a more practical perspective one can try to show that large classes of problems that are arguably ‘natural’ enjoy this dichotomy property. This approach has been suggested for several classes of problems, most notably by Valiant in the context of counting problems, the Holant problem and holographic algorithms [Valiant 2006; 2008], and by Feder and Vardi in the context of the Constraint Satisfaction Problem (CSP) [Feder and Vardi 1993; 1998]. Since this paper is about the CSP, we discuss the latter research direction in more details.

In [Feder and Vardi 1993; 1998] Feder and Vardi start off by a systematic study of problems representable through logic formulas that may exhibit the dichotomy phenomenon. Due to Fagin’s theorem we know that every problem in NP can be represented as deciding if the input structure satisfies a second-order existential formula. So, Feder and Vardi tried to identify the largest class of formulas that does not represent the whole of NP, and therefore may have the dichotomy property. The candidate class they arrived at was MMSNP, the class of monotone monadic formulas without inequalities. In particular, they show that if any of these three properties is eliminated,
any problem from NP is polynomial time equivalent to a problem in the resulting class, and therefore this class cannot have the dichotomy property. On the other hand, they show that for every problem from the class MMSNP itself there is a randomized polynomial time reduction to a certain nonuniform CSP\(^2\). This led them to the following CSP Dichotomy Conjecture

**Conjecture 1.1 (CSP Dichotomy Conjecture, [Feder and Vardi 1993; 1998]).**

*For every finite relational structure \(\mathcal{H}\), the nonuniform constraint satisfaction problem, \(\text{CSP}(\mathcal{H})\), either can be solved in polynomial time, or is NP-complete.***

In this paper we discuss how the CSP Dichotomy Conjecture has been resolved, starting from early dichotomy results of Schaefer [Schaefer 1978] on the Generalized Satisfiability Problem, and Hell and Nešetřil [Hell and Nešetřil 1990] on Graph H-Colouring, to its final resolution by the author [Bulatov 2017b; 2017c] and independently and nearly simultaneously by Zhuk [Zhuk 2017a; 2018]. Most of the paper uses the links between the CSP and universal algebra — the approach that has been very effective in the study of the CSP. However, before turning to algebraic concepts we touch upon other approaches and provide a number of results and examples obtained there.

Complexity classification of the decision CSP and its variants has been moved beyond the Dichotomy conjecture. Dichotomy or other complexity classification results have been proved (or sometimes conjectured) for a number of variants of the CSP. We will mention some of these results towards the end of the paper.

2. THE CONSTRAINT SATISFACTION PROBLEM

We begin with defining the CSP in a way that historically has been used by researchers in Artificial Intelligence, and will be convenient in this paper. In the definition below tuples of elements are denoted in boldface, say, \(a\), and the \(i\)th component of \(a\) is referred to as \(a[i]\). The set \(\{1, \ldots, n\}\) will be denoted by \([n]\).

**Definition 2.1.** Let \(A_1, \ldots, A_\ell\) be finite sets. An instance \(I = (V, C)\) of the CSP over \(A_1, \ldots, A_\ell\) consists of a finite set of variables \(V\) such that each \(v \in V\) is assigned a domain \(A_{i_v}\), \(i_v \in [\ell]\), and a finite set \(C\) of constraints. Each constraint is a pair \((s, R)\) where \(R\) is a relation over \(A_1, \ldots, A_\ell\) (say, \(k\)-ary), often called the *constraint relation*, and \(s\) is a \(k\)-tuple of variables from \(V\), called the *constraint scope*. Let \(\sigma : V \rightarrow A = A_1 \cup \cdots \cup A_\ell\) be a mapping with \(\sigma(v) \in A_{i_v}\); we write \(\sigma(s)\), for \((\sigma(s[1]), \ldots, \sigma(s[k]))\). A solution of \(I\) is a mapping \(\sigma : V \rightarrow A\) such that for every constraint \((s, R) \in C\) we have \(\sigma(s) \in R\). The objective in the CSP is to decide whether or not a solution of a given instance \(I\) exists.

Since its inception in the early 70s [Mackworth 1977], the CSP has become a very popular and powerful framework, widely used to model computational problems first in artificial intelligence, [Dechter 2003] and later in many other areas.

Restrictions of the general CSP similar to nonuniform CSPs from the previous section can be introduced through constraint languages. Let \(A_1, \ldots, A_\ell\) be finite sets and \(\Gamma\) a set (finite or infinite) of relations over \(A_1, \ldots, A_\ell\), called a *constraint language*. Then \(\text{CSP}(\Gamma)\) is the class of all instances \(I\) of the CSP such that \(R \in \Gamma\) for every constraint \((s, R)\) from \(I\). The following examples are just a few of the problems representable as \(\text{CSP}(\Gamma)\).

**Example 2.2.** *(k-Col)* The standard \(k\)-Colouring problem has the form \(\text{CSP}(\Gamma_{\text{kCol}})\), where \(\Gamma_{\text{kCol}} = \{\neq_k\}\) and \(\neq_k\) is the disequality relation on a \(k\)-element

\(^2\)These reductions have been derandomized by Kun [Kun 2013].
set (of colours).

(3-SAT) An instance of the 3-SAT problem is a propositional logic formula in CNF each clause of which contains 3 literals, and the goal is to decide if it has a satisfying assignment. Thus, 3-SAT is equivalent to CSP(Γ_3SAT), where Γ_3SAT is the constraint language on \{0, 1\} that contains relations R_1, ..., R_8, which are the 8 ternary relations that can be expressed by a 3-clause.

(LIN) Let \( F \) be a finite field and let LIN(\( F \)) be the problem of deciding the consistency of a system of linear equations over \( F \). Then LIN(\( F \)) is equivalent to CSP(Γ_{LIN(\( F \))}), where Γ_{LIN(\( F \))} is the constraint language over \( F \) whose relations are given by a linear equation.

As the examples above indicate, in most cases our CSPs start off as problems over a single domain. However, solution algorithms often modify the domains of variables in different ways.

We illustrate the correspondence between the homomorphism definition of the CSP and Definition 2.1 with an example. Consider again the \( k \)-COLOURING problem, and let \( \mathcal{H}_k \) denote the relational structure with universe \( \{k\} \) over vocabulary \( \{R_6\} \) and \( R^H_6 \) is interpreted as the disequality relation. In other words, \( \mathcal{H}_k = K_k \) is a complete graph with \( k \) vertices. Then a homomorphism from a given graph \( G = (V, E) \) to \( K_k \) exists if and only if it is possible to assign vertices of \( K_k \) (colours) to vertices of \( G \) in such a way that for any \( (u, v) \in E \) the vertices \( u \) and \( v \) are assigned different colours. The latter is just a proper \( k \)-colouring of \( G \).

Using the homomorphism definition the \( k \)-COLOURING problem can be generalized to the \( H \)-COLOURING problem, where \( H \) is a graph or digraph: Given a (di)graph \( G \), decide whether or not there is a homomorphism from \( G \) to \( H \). Using the CSP notation the \( H \)-COLOURING problem is CSP(\( E_H \)), where \( E_H \) denotes the edge relation of \( H \). The \( H \)-COLOURING problem has received much attention in graph theory, see, e.g. [Hell and Nešetřil 2004; Hell and Nešetřil 1990].

3. LOGIC AND CONSTRAINT PROPAGATION

3.1. Logic and Databases

The next step in the CSP research was motivated by its applications in the theory of relational databases. The QUERY EVALUATION problem can be thought of as deciding whether a first order sentence in the vocabulary of a database is true in that database (that is, whether or not the query has a non-empty answer). The QUERY CONTAINMENT problem asks, given two queries \( \Phi \) and \( \Psi \), whether the query \( \Phi \to \Psi \) is true in all databases of the given vocabulary. The former problem is of course the main problem relational databases are needed for, while the latter is routinely used in various query optimization techniques. It turns out that both problems have intimate connections to the CSP, if the CSP is properly reformulated. We need some terminology from model theory.

A vocabulary is a finite set of relational symbols \( R_1, \ldots, R_n \) each of which has a fixed arity \( \text{ar}(R_i) \). A relational structure over vocabulary \( R_1, \ldots, R_n \) is a tuple \( \mathcal{H} = (H; R^H_1, \ldots, R^H_n) \) such that \( H \) is a non-empty set, called the universe of \( \mathcal{H} \), and each \( R^H_i \) is a relation over \( H \) having the same arity as the symbol \( R_i \). A sentence is said to be a conjunctive query if it only uses existential quantifiers and its quantifier-free part is a conjunction of atomic formulas.
Definition 3.1. An instance of the CSP is a pair \((\Phi, \mathcal{H})\), where \(\mathcal{H}\) is a relational structure in a certain vocabulary, and \(\Phi\) is a conjunctive query in the same vocabulary. The objective is to decide whether \(\Phi\) is true in \(\mathcal{H}\).

To see that the definition above is equivalent to the previous two definitions of the CSP, we again consider its special case, \(k\)-\textsc{COLOURING}. The vocabulary corresponding to the problem contains just one binary predicate \(R_\neq\). Let again \(\mathcal{H}_k\) be the relational structure with universe \([k]\) in the vocabulary \(\{R_\neq\}\), where \(R_\neq^{\mathcal{H}_k}\) is interpreted as the disequality relation on the set \([k]\). Then an instance \(G = ([v_1, \ldots, v_n], E)\) of \(k\)-\textsc{COLOURING} is equivalent to testing whether conjunctive query
\[\exists x_1, \ldots, x_n \land_{(v_i, v_j) \in E} R_\neq(x_i, x_j)\]
is true in \(\mathcal{H}\).

Thus, the \textsc{Query Evaluation} problem, when restricted to conjunctive queries, is just the CSP. A database is then considered as the input relational structure. The Chandra-Merlin Theorem [Chandra and Merlin 1977] shows that the \textsc{Query Containment} problem is also equivalent to the CSP.

Relational database theory also massively contributed to the CSP research, most notably by techniques related to local propagation algorithms and the logic language Datalog that we will discuss next.

3.2. Local Propagation Algorithms

Constraint propagation algorithms are probably the most natural way to solve a CSP, and have been intensively studied and widely used in AI since the very beginning. There is a variety of such algorithms (see [Dechter 2003] to have some idea) differing in strength and running time, but we describe essentially one such algorithm, applicable whenever any other propagation algorithm solves the problem.

Let \(R \subseteq A_1 \times \cdots \times A_k\) be a relation, \(a \in A_1 \times \cdots \times A_k\), and \(J = \{i_1, \ldots, i_k\} \subseteq [\ell]\). Let \(pr_j a = [a[i_1], \ldots, a[i_k]]\) and \(pr_j R = \{pr_j a \mid a \in R\}\), be the projections of \(a\) and \(R\) respectively, on \(J\). Often we will use sets of CSP variables to index entries of tuples and relations. Projections in this case are defined in a similar way. Let \(I = (V,C)\) be a CSP instance. For \(W \subseteq V\) by \(I_W\) we denote the restriction of \(I\) onto \(W\), that is, the instance \((W, C_W)\), where for each \(C = (s, R) \in C\), the set \(C_W\) includes the constraint \(C_W = (s \cap W, pr_{W,R})\). The set of solutions of \(I_W\) will be denoted by \(S_W\).

Unary solutions, that is, when \(|W| = 1\) play a special role. As is easily seen, for \(v \in V\) the set \(S_v\) is just the intersection of unary projections \(pr_v R\) of constraints whose scope contains \(v\). Instance \(I\) is said to be \(1\)-\textit{minimal} if for every \(v \in V\) and every constraint \(C = (s, R) \in C\) such that \(v \in s\), it holds \(pr_v R = S_v\). For a 1-minimal instance one may always assume that allowed values for a variable \(v \in V\) is the set \(S_v\). We call this set the domain of \(v\). The domain \(S_v\) may change as a result of transformations of the instance.

Instance \(I\) is said to be \((2,3)\)-\textit{minimal} if it satisfies the following condition:
- for every \(X = \{u, v\} \subseteq V\), any \(w \in V - X\), and any \((a, b) \in S_X\), there is \(c \in S_w\) such that \((a, c) \in S_{\{u,w\}}\) and \((b, c) \in S_{\{v,w\}}\).

For \(k \in \mathbb{N}\), \((k, k+1)\)-minimality is defined in a similar way using \(k, k+1\) instead of 2,3.

Instance \(I\) is said to be \textit{minimal} (or \textit{globally minimal}) if for every \(C = (s, R) \in C\) and every \(a \in R\) there is a solution \(\varphi\) such that \(\varphi(s) = a\). Similarly, \(I\) is said to be \textit{globally 1-minimal} if for every \(v \in V\) and \(a \in S_v\) there is a solution \(\varphi\) with \(\varphi(v) = a\). Clearly, establishing minimality amounts to solving the problem and so not always can be easily done.

Any instance can be transformed to a 1-minimal or \((2,3)\)-minimal instance in polynomial time using the standard constraint propagation algorithms (see, e.g. [Dechter...
These algorithms work by changing the constraint relations and the domains of the variables eliminating some tuples and elements from them.

If a constraint propagation algorithm solves a CSP, the problem is said to be of bounded width. More precisely, $\text{CSP}(\Gamma)$ is said to have bounded width if for some $k$ every $(k,k+1)$-minimal instance from $\text{CSP}(\Gamma)$ has a solution (we also say that $\text{CSP}(\Gamma)$ has width $k$ in this case).

**Example 3.2.**

1. The 2-SAT problem has bounded width, namely, width 2.

2. The $H$-COLOURING problem has width 2 when graph $H$ is bipartite, and is NP-complete otherwise.

3. HORN-SAT is the SATISFIABILITY problem restricted to Horn clauses, i.e., clauses of the form $x_1 \land \cdots \land x_k \rightarrow y$. Let $\Gamma_{k,HORN}$ be the constraint language consisting of relations expressible by a Horn clause with at most $k$ premises. The problem $k$-HORN-SAT is equivalent to $\text{CSP}(\Gamma_{k,HORN})$ and has width $k$.

4. The LIN problem provides an archetypical example of a CSP that does not have bounded width. Indeed, for any $k$ it is not difficult to construct a system of linear equations that is inconsistent, but the $(k,k+1)$-minimality algorithm returns an instance with non-empty constraints.

Problems of bounded width are well studied, see the older survey [Bulatov et al. 2008] and more recent [Barto 2014a]. Barto and Kozik [Barto and Kozik 2014; Barto 2014a] and independently Bulatov [Bulatov 2016c] characterized languages $\Gamma$ such that $\text{CSP}(\Gamma)$ has bounded width. As this characterization involves some algebraic concepts, we return to it later.

### 3.3. Datalog and other equivalent characterizations of bounded width

Problems of bounded width have several equivalent characterizations.

**Datalog.** This simple language is related to the least fixed point operator in logic and uses the predicates of some relational structure as well as certain derived predicates. For example, let $G = (V,E)$ be a graph, and let us consider the following simple Datalog program

\[
\begin{align*}
P(x,y) & \quad \leftarrow \quad E(x,y) \\
P(x,y) & \quad \leftarrow \quad P(x,z) \land E(z,t) \land E(t,y) \\
O & \quad \leftarrow \quad P(x,x)
\end{align*}
\]

Here, say, the second rule (line) means that predicate $P$ has to be true on all pairs $x,y$, for which there are $z,t$ satisfying the expression on the right hand side. It is easy to see that predicate $O$ here becomes true if and only if $G$ contains an odd cycle. In this sense the Datalog program above decides whether a given graph is NOT bipartite. For more basics of Datalog and its applications see [Kolaitis and Vardi 1995; Kolaitis 2007].

**Homomorphism duality.** Let $\mathcal{H}$ be a structure, a set $O$ is said to be an obstruction set for $\mathcal{H}$ if for any structure $G$, there is a homomorphism from $G$ to $\mathcal{H}$ if and only if for no structure $G' \in O$ there is a homomorphism from $G'$ to $G$. For instance, the set of odd cycles is an obstruction set for any bipartite graph. Structure $\mathcal{H}$ is said to have tree-width $k$ duality if there is an obstruction set $O$ for $\mathcal{H}$ such that the tree-width of every structure of $O$ is at most $k$. Thus, any bipartite graph has tree-width 2 duality. Homomorphism duality originates in graph theory [Hell and Nešetřil 2004], and has been used for CSP as well.
Pebble games. The existential \( k \)-pebble game has been introduced in [Kolaitis and Vardi 1995]. It is played on a pair of structures \( G \) and \( H \) by two players \textit{Spoiler} and \textit{Duplicator}. Spoiler has \( k \) pebbles and in every move she can place or remove a pebble on an element of \( G \). Duplicator has to respond by maintaining a partial homomorphism from \( G \) to \( H \). More precisely, if \( I \subseteq G \) is the current set of pebbled elements of \( G \) and \( \varphi : I \to H \) is the current partial homomorphism, then, if Spoiler removes a pebble, Duplicator has to respond with the corresponding restriction of \( \varphi \). If Spoiler adds a pebble, Duplicator has to respond with an extension of \( \varphi \). Spoiler wins if at any point of time Duplicator is unable to respond. Duplicator wins if she has a strategy of keeping the game going forever.

It turns out that the three frameworks above lead to the same class(es) of the CSP.

\textbf{Theorem 3.3} ([Kolaitis and Vardi 1995; Feder and Vardi 1998]). For a finite relational structure \( H \) and \( k \in \mathbb{N} \) the following conditions are equivalent:

(1) There is a Datalog program \( \mathcal{P} \) whose rules use no more than \( k \) different variables that defines the class of structures not homomorphic to \( H \).

(2) \( H \) has tree-width \( k \) duality.

(3) A structure \( G \) is homomorphic to \( H \) if and only if Duplicator has a winning strategy in the \( k \)-pebble game on \( G, H \).

Although CSPs of width \( k \) (as defined here) are not captured by the conditions of Theorem 3.3, those conditions capture bounded width.

\textbf{Theorem 3.4} ([Barto and Kozik 2012; Barto 2014a; Bulatov 2016c]). For a finite relational structure \( H \) the following conditions are equivalent:

(1) \( \text{CSP}(H) \) has width 2, that is, establishing (2,3)-minimality is a solution algorithm.

(2) \( \text{CSP}(H) \) has width \( \ell \) for some \( \ell \geq 2 \), that is, \( \text{CSP}(H) \) has bounded width.

(3) there is \( k \), for which the equivalent conditions of Theorem 3.3 hold.

From the practical perspective establishing (2,3)-minimality, and moreover \((k,k+1)\)-minimality for \( k > 2 \) is very computationally demanding. It is therefore important to find the fastest propagation algorithm that nevertheless solves problems of bounded width. So far the fastest such algorithm was suggested by Kozik [Kozik 2016].

Note that if a relational structure \( H \) has infinite number of predicates or, equivalently, a constraint language is infinite, Theorems 3.3.3.4 are no longer true, and some of the conditions (such as definability by Datalog) are even not applicable. The concept of width and bounded width of a CSP nevertheless remains valid, as well as, the equivalence of conditions (1),(2) of Theorem 3.4.

\textbf{4. ALGEBRAIC APPROACH}

The most successful approach to tackling the Dichotomy Conjecture turned out to be the algebraic one. In this section we introduce the algebraic approach to the CSP and show how it can be used to determine the complexity of nonuniform CSPs. A keen reader can find more details on the algebraic approach, its applications, and the underlying algebraic facts from the following books [Grätzer 2008; Hobby and McKenzie 1988], surveys [Barto et al. 2017; Barto and Kozik 2017; Bulatov and Valeriote 2008; Bulatov et al. 2008], and research papers [Bulatov et al. 2005; Bulatov 2006b; 2011; 2016a; Bulatov and Dalmau 2006; Berman et al. 2010; Barto 2014a; Barto and Kozik 2014; 2012; Idziak et al. 2010].
4.1. Primitive Positive Definitions

Let \( \Gamma \) be a set of relations (predicates) over a finite set \( A \). A relation \( R \) over \( A \) is said to be primitive-positive (pp-) definable in \( \Gamma \) if \( R(x) = \exists y \Phi(x,y) \), where \( \Phi \) is a conjunction that involves predicates from \( \Gamma \) and equality relations. The formula above is then called a pp-definition of \( R \) in \( \Gamma \). A constraint language \( \Delta \) is pp-definable in \( \Gamma \) if so is every relation from \( \Delta \). In a similar way pp-definability can be introduced for relational structures.

**Example 4.1.** Let \( K_3 = ([3], E) \) be a 3-element complete graph. Its edge relation is the binary disequality relation on \( [3] = \{1, 2, 3\} \). Then the pp-formula

\[
Q(x, y, z) = \exists t, u, v, w (E(t, x) \wedge E(t, y) \wedge E(t, z) \wedge E(u, v) \wedge E(v, w) \\
\wedge E(w, u) \wedge E(u, x) \wedge E(v, y) \wedge E(w, z))
\]

defines the relation \( Q \) that consists of all triples containing exactly 2 different elements from \([3]\).

A link between pp-definability and reducibility between nonuniform CSPs was first observed by Jeavons et al. in [Jeavons et al. 1997].

**Theorem 4.2** ([Jeavons et al. 1997]). Let \( \Gamma \) and \( \Delta \) be constraint languages and \( \Delta \) finite. If \( \Delta \) is pp-definable in \( \Gamma \) then \( \text{CSP}(\Delta) \) is polynomial time reducible\(^3\) to \( \text{CSP}(\Gamma) \).

It was later shown that pp-definability in Theorem 4.2 can be replaced with a more general notion of pp-constructability [Barto et al. 2017; Barto et al. 2018].

4.2. Polymorphisms and Invariants

Primitive positive definability can be concisely characterized using polymorphisms. An operation \( f : A^k \to A \) is said to be a polymorphism of a relation \( R \subseteq A^n \) if for any \( a_1, \ldots, a_k \in R \) the tuple \( f(a_1, \ldots, a_k) \) also belongs to \( R \), where \( f(a_1, \ldots, a_k) \) stands for \( (f(a_1[1], \ldots, a_k[1]), \ldots, f(a_1[n], \ldots, a_k[n])) \). Operation \( f \) is a polymorphism of a constraint language \( \Gamma \) if it is a polymorphism of every relation from \( \Gamma \). Similarly, operation \( f \) is a polymorphism of a relational structure \( H \) if it is a polymorphism of every relation of \( H \). The set of all polymorphisms of language \( \Gamma \) or relational structure \( H \) is denoted by \( \text{Pol}(\Gamma) \), \( \text{Pol}(H) \). If \( F \) is a set of operations, \( \text{Inv}(F) \) denotes the set of all relations \( R \) such that every operation from \( F \) is a polymorphism of \( R \).

**Example 4.3.** Let \( R \) be an affine relation, that is, \( R \) is the solution space of a system of linear equations over a field \( F \). Then the operation \( f(x, y, z) = x - y + z \) is a polymorphism of \( R \). Indeed, let \( A \cdot x = b \) be a system defining \( R \), and \( x, y, z \in R \). Then

\[
A \cdot f(x, y, z) = A \cdot (x - y + z) = A \cdot x - A \cdot y + A \cdot z = b - b + b = b.
\]

In fact, the converse can also be shown: if \( R \) is invariant under \( f \), where \( f \) is defined in a certain finite field \( F \) then \( R \) is the solution space of some system of linear equations over \( F \).

Several other useful polymorphisms are the following [Jeavons et al. 1997; Jeavons et al. 1998; Bulatov and Dalmu 2006; Maróti and McKenzie 2008]

**Example 4.4.** (1) A binary operation \( \cdot \) on a set \( A \) is said to be a semilattice operation if it is (a) idempotent, \( x \cdot x = x \); (b) commutative, \( x \cdot y = y \cdot x \); and (c) associative, \( x \cdot (y \cdot z) = (x \cdot y) \cdot z \), for any \( x, y, z \in A \).

\(^3\)In fact, due to the result of [Reingold 2008] this reduction can be made log-space.
(2) A \( k \)-ary operation \( g \) on \( A \) is called a near-unanimity operation, or NU if
\[
g(y, x, \ldots, x) = g(x, y, x, \ldots, x) = \cdots = g(x, \ldots, x, y) = x
\]
for any \( x, y \in A \). A ternary NU is also referred to as a majority operation.

(3) A \( k \)-ary operation \( g \) on \( A \) is called a weak near-unanimity operation, or WNU if it satisfies all the equations of an NU except for the last one
\[
g(y, x, \ldots, x) = g(x, y, x, \ldots, x) = \cdots = g(x, \ldots, x, y).
\]

(4) A ternary operation \( h \) on \( A \) is called Maltsev if
\[
h(x, y, y) = h(y, y, x) = x
\]
for any \( x, y \in A \). As we saw in Example 4.3 any structure whose relations can be represented by linear equations has the Maltsev polymorphism \( x - y + z \) where + and – are the operations of the underlying field.

A link between polymorphisms and pp-definability of relations is given by the Galois connection.

**THEOREM 4.5 (GALOIS CONNECTION, [BODNARUCH ET AL. 1969; GEIGER 1968]).**

Let \( \Gamma \) be a constraint language on \( A \), and let \( R \subseteq A^n \) be a non-empty relation. Then \( R \) is preserved by all polymorphisms of \( \Gamma \) if and only if \( R \) is pp-definable in \( \Gamma \).

Theorems 4.2 and 4.5 together imply that the complexity of CSP(\( \Gamma \)) depends entirely on Pol(\( \Gamma \)). This was used by Jeavons and coauthors in the early papers on the algebraic approach to show the tractability and hardness of certain classes of the CSP.

**Example 4.6.** (A) If a constraint language \( \Gamma \) has a semilattice or NU polymorphism, then CSP(\( \Gamma \)) can be solved in polynomial time, [Jeavons et al. 1997; Jeavons et al. 1998].

(B) If \( \Gamma \) has a Maltsev polymorphism \( h \), then CSP(\( \Gamma \)) can be solved in polynomial time, [Bulatov 2002b; Bulatov and Dalmau 2006]. In certain special cases — when \( h \) is an affine operation or the operation \( xy^{-1}z \) of a finite group — this result can be traced back to [Feder and Vardi 1998; Jeavons et al. 1997]. However, the general result turns out to be much more difficult than those from the previous item.

(C) If every polymorphism \( f \) of a constraint language \( \Gamma \) on a set \( A \) is such that \( f(x_1, \ldots, x_n) = x_i \) for some \( i \) and all \( x_1, \ldots, x_n \in A \), then CSP(\( \Gamma \)) is NP-complete [Jeavons et al. 1997].

(D) Schaefer’s Theorem [Schaefer 1978] can be stated in terms of polymorphisms. Let \( \Gamma \) be a constraint language on a 2-element set (we assume this set to be \( \{0, 1\} \)). The problem CSP(\( \Gamma \)) is solvable in polynomial time if and only if one of the following operations is a polymorphism of \( \Gamma \): the constant operations 0 or 1, the semilattice operations of conjunction and disjunction, the majority operation on \( \{0, 1\} \) (there is only one such operation), or the Maltsev operation \( x - y + z \) where + and – are modulo 2. Otherwise CSP(\( \Gamma \)) is NP-complete.

### 4.3. Algebras and the CSP

Recall that a (universal) algebra is an ordered pair \( A = (A, F) \) where \( A \) is a non-empty set, called the universe of \( A \), and \( F \) is a family of finitary operations on \( A \), called the basic operations of \( A \). Operations that can be obtained from \( F \) by means of composition are said to be term operations of the algebra. Every constraint language on a set \( A \) can be associated with an algebra \( \text{Alg}(\Gamma) = (A, \text{Pol}(\Gamma)) \). In a similar way any relational structure \( A \) (with universe \( A \)) can be paired up with the algebra \( \text{Alg}(A) = (A, \text{Pol}(A)) \). On the other hand, an algebra \( A = (A, F) \), can be associated with
the constraint language $\text{Inv}(F)$ or the class $\text{Str}(A)$ of structures $A = (A, R_1, \ldots, R_k)$ such that $R_1, \ldots, R_k \in \text{Inv}(F)$.

This correspondence can be extended to CSPs: For an algebra $A$ by $\text{CSP}(A)$ we denote the class of problems $\text{CSP}(A), A \in \text{Str}(A)$. Equivalently, $\text{CSP}(\text{Inv}(F))$ can be thought of as $\text{CSP}(\text{Inv}(F))$ for the infinite constraint language $\text{Inv}(F)$. Note, however, that there is a subtle difference in the notion of polynomial time solvability in these two cases that we will address next.

We say that algebra $A$ is \textit{tractable} if every $\text{CSP}(A), A \in \text{Str}(A)$, is solvable in polynomial time. Observe that this does not guarantee that there is a single solution algorithm for all such problems, nor does it guarantee that there is any uniformity among those algorithms. In general, it may be possible that for a tractable algebra $A = (A, F)$ the problem $\text{CSP}(\text{Inv}(F))$ is NP-hard. If the problem $\text{CSP}(\text{Inv}(F))$ is solvable in polynomial time, we call $A$ \textit{globally tractable}. Algebra $A$ is called $\textit{NP-complete}$ if some $\text{CSP}(A), A \in \text{Str}(A)$, is NP-complete. Algebra $A$ is \textit{globally NP-complete} if $\text{CSP}(\text{Inv}(F))$ is NP-complete.

Using the algebraic terminology we can pose a stronger version of the Dichotomy Conjecture.

\textbf{Conjecture 4.7 (Dichotomy Conjecture+).} Every finite algebra either is \textit{globally tractable} or is \textit{NP-complete} (in the local sense).

The Dichotomy Conjecture+ can be made more precise by making use of weak near-unanimity terms. An operation $f$ on a set $A$ is said to be \textit{idempotent} if the equality $f(x, \ldots, x) = x$ holds for all $x \in A$. An algebra all of whose term operations are idempotent is said to be \textit{idempotent}.

\textbf{Theorem 4.8 ([Bulatov et al. 2005]).} For any finite algebra $A$ there is an idempotent finite algebra $B$ such that:

- $A$ is globally tractable if and only if $B$ is globally tractable;
- $A$ is NP-complete if and only if $B$ is NP-complete.

Theorem 4.8 reduces the Dichotomy Conjecture+ to idempotent algebras.

\textbf{Conjecture 4.9.} If a relational structure $A$ is such that $\text{Alg}(A)$ is idempotent, then $\text{CSP}(A)$ is solvable in polynomial time if and only if $A$ admits a weak near-unanimity polymorphism. Otherwise it is NP-complete.

Or in the stronger algebraic version

\textbf{Conjecture 4.10 (Dichotomy Conjecture ++).} An idempotent algebra $A$ is globally tractable if and only if it has a weak near-unanimity term operation. Otherwise it is NP-complete.

By the results of [Maróti and McKenzie 2008] the Dichotomy Conjecture ++ is equivalent to the conjecture stated in [Bulatov et al. 2005].

5. THE PURSUIT OF THE DICHOTOMY CONJECTURE

The complexity of the CSP including the dichotomy conjecture has been intensively studied for 40 years. Here we outline the history of this area that is related to the dichotomy conjecture.

\textit{Early dichotomies.} Schaefer [Schaefer 1978] obtained the first dichotomy theorem on the CSP, long before the Dichotomy Conjecture was proposed. His classification of constraint languages on a 2-element set can be easily extended to 2-element algebras. Then it claims that an idempotent 2-element algebra is globally tractable if and only
if it has one of the following term operations: a semilattice operation, a majority operation, or the affine operation $x - y + z$. By [Post 1941] this is equivalent to having a weak near-unanimity term operation. Another early dichotomy result by Hell and Nesetril [Hell and Nešetřil 1990] gives a classification of (undirected) graphs $H$ with respect to the complexity of the $H$-COLOURING problem: such a problem is polynomial time solvable if $H$ is bipartite or contains a loop, and NP-complete otherwise. Let $H$ be a graph, $\mathcal{A} = \text{Alg}(H)$, and let $\mathcal{B}$ be the idempotent algebra constructed from $\mathcal{A}$ as in Theorem 4.8. If $H$ is bipartite then $\mathcal{B}$ is 2-element and has a majority term operation. Otherwise $\mathcal{B}$ does not have a WNU [Bulatov 2005]. Thus the classification from [Hell and Nešetřil 1990] matches the Dichotomy Conjecture++.

The two algorithms. Apart from posing the Dichotomy Conjecture Feder and Vardi [Feder and Vardi 1998] made several important observations. One of them is a clear distinction between two types of CSP algorithms. One might expect that problems as diverse as nonuniform CSPs would have a variety of different solution algorithms. This, however, is not the case, and all known (at that point) algorithms are variations of just two types of algorithms. Algorithms of the first type include all local propagation algorithms as described in Section 3.2. Feder and Vardi conjectured that the solvability of $\text{CSP}(\mathcal{H})$ by a local propagation algorithm is related to a property they called the ‘ability to count’. More precisely, they conjectured that a propagation algorithm solves $\text{CSP}(\mathcal{H})$ if and only if $\mathcal{H}$ does not have the ability to count. This conjecture has been confirmed much later in [Larose et al. 2009] (if combined with the characterization of CSPs of bounded width [Barto and Kozik 2014; Bulatov 2016c]). Algorithms of the second type use variations of Gaussian Elimination or group theoretic approaches such as Furst’s algorithm for coset generation [Furst et al. 1980] which is used in [Feder and Vardi 1998].

Algebraic approach, polymorphisms. The discovery by Jeavons et al. [Jeavons et al. 1997; Jeavons et al. 1998] of the connection between polymorphisms and the complexity of the CSP allowed, firstly, to greater unify approaches to different constraint languages. It turned out, many of them have polymorphisms possessing similar properties and so could be handled in similar ways. In particular, the fact that all the assorted constraint languages known to have bounded width actually have this property can be explained by polymorphisms of just two types [Jeavons et al. 1998]: NU and semilattice operations. In the former case bounded width follows from the decomposition theorem by Pixley [Pixley 1979], while in the latter case bounded width is an easy implication of the structure of semilattices. Similarly, in nearly all other cases the tractability of a CSP could be explained by the existence of a group Mal’tsev polymorphism $xy^{-1}z$ or $x - y + z$. For a recent survey of the algebraic approach see [Barto et al. 2017].

Algebraic approach: algebras and varieties. Extending the algebraic approach from polymorphisms to algebras and varieties of algebras [Bulatov et al. 2005; Bulatov and Jeavons 2001; 2003] contributed to the study of the CSP in two ways. Firstly, it demonstrated that what is important for the complexity of CSPs is not particular polymorphisms, but the identities or equations they satisfy (cf. Example 4.4). Secondly, it allowed to employ various structural theories of universal algebras. In particular, it made possible dichotomy results on arbitrary CSPs on small domains, they all agree with the Dichotomy Conjecture++, [Bulatov 2002a; 2006b] (3-element domains), [Marković 2011] (4-element domains), [Zhuk 2016b; 2016a] (5- and 7-element domains). Although still somewhat ad-hoc and based on case analysis, such results had been inaccessible by previous methods for more than 20 years. This structural approach also made it possible to design algorithms for several types of polymorphisms: Mal’tsev polymorphisms, see Example 4.4(4) ([Bulatov 2002b; Bulatov and Dalmau...
2006] subsequently generalized in [Dalmau 2006]), and 2-semilattice polymorphisms satisfying the identities $xx = x$, $xy = yx$, $x(yx) = (xy)x$, [Bulatov 2006a]. Another direction that became possible due to the algebraic approach is the finer classification of complexity of the CSP. Initiated in [Allender et al. 2009] it was later developed in [Larose and Tesson 2009], where a conjecture was posed on such a classification using the language of omitting types in the sense of tame congruence theory [Hobby and McKenzie 1988], and then in [Larose et al. 2007; Kazda 2018].

**Absorption and bounded width.** Interaction with the CSP research catalyzed the development of universal algebra as well, since new tools were needed. One of the major developments was the concept of absorption and related techniques introduced by Barto and Kozik [Barto and Kozik 2012; Barto 2014b], see also [Barto and Kozik 2017] for a recent survey. Fundamental technical lemmas obtained in those papers, such as the Loop Lemma and the Rectangularity Lemma, allowed them to prove a number of major results in universal algebra. For CSP they led to a dichotomy theorem for digraphs without sources and sinks [Barto et al. 2009] and some other classes of digraphs [Barto et al. 2009; Barto and Bulin 2013]. The most important result obtained using this technique is the characterization of problems of bounded width.

**Theorem 5.1** ([Barto 2014A; Bulatov 2016C; 2004; Kozik et al. 2015]).
For an idempotent algebra $\Lambda$ the following are equivalent:
(1) $\text{CSP}(\Lambda)$ has bounded width;
(2) every (2,3)-minimal instance from $\text{CSP}(\Lambda)$ has a solution;
(3) $\Lambda$ has a weak near-unanimity term operation of arity $k$ for every $k \geq 3$;
(4) every quotient algebra of a subalgebra of $\Lambda$ has a nontrivial operation, and none of them is equivalent to a module (in a certain precise sense).

**Few subpowers algorithm.** The second type of algorithms identified in [Feder and Vardi 1998] is based on group theoretic tools and has been generalized to problems with a Mal’tsev polymorphism [Bulatov 2002b; Bulatov and Dalmau 2006], and then to problems with a Generalized Majority-Minority polymorphism [Dalmau 2006]. The common feature of all those algorithms was that similar to Gaussian Elimination they construct some sort of a compact basis of the set of solutions. Such a basis may not exist in the general case.

It is thought that the property of relations to have a compact representation, where compactness is understood as having size polynomial in the arity of the relation, is the right generalization of linear algebra problems where Gaussian Elimination can be used. Let $\Lambda = (A, F)$ be an algebra. It is said to be an algebra with few subpowers if every relation over $A$ invariant under $F$ admits a compact representation [Berman et al. 2010; Idziak et al. 2010]. The term few subpowers comes from the observation that every relation invariant under $F$ is a subalgebra of a direct power of $\Lambda$, and if the size of compact representation is bounded by a polynomial $p(n)$ then at most $2^{p(n)}$ $n$-ary relations can be represented, while the total number of such relations can be as large as $2^{|A|^n}$. Algebras with few subpowers are completely characterized by Idziak et al. [Berman et al. 2010; Idziak et al. 2010]. A minor generalization of the algorithm from [Dalmau 2006] solves CSP($\Lambda$), where $\Lambda$ has few subpowers.

**Conservative CSPs and graphs of algebras.** An important general class of constraint languages where a dichotomy theorem could be obtained by more or less ad-hoc methods is the class of conservative languages: A language $\Gamma$ over a set $A$ is said to be conservative if every subset of $A$ is a (unary) relation in $\Gamma$. In terms of the CSP it means that in an instance the set of possible values of each variable can be arbitrarily restricted. Similar problems have been studied within the graph homomorphism
community, where they are called **LIST HOMOMORPHISM PROBLEMS** (as every vertex of the source graph is equipped with a list of possible images), see [Hell and Nešetřil 2004; Feder and Hell 1998; Feder et al. 1999]. As is shown in [Bulatov 2011; 2003] the Dichotomy Conjecture++ holds for such constraint languages. This result was later greatly simplified in [Barto 2011; Bulatov 2016a]. The main approach used in these proofs (except [Barto 2011] that is based on absorption) has later proved to be the key to resolving the dichotomy conjecture. For any polynomial time solvable conservative \( \Gamma \) on a set \( A \) the problem \( \text{CSP}(\Gamma) \) restricted to a 2-element subset of \( A \) is equivalent to one of Schaefer’s cases and therefore has one of the good polymorphisms: semilattice, majority, or affine. Thus within this approach 2-element subsets of \( A \) are assigned types that depend on which good polymorphism occurs in the restricted problem, converting \( A \) into an edge-coloured graph. This approach has been further generalized for arbitrary constraint languages (algebras) in [Bulatov 2004; 2016b; 2016c].

**Hybrid algorithms.** CSPs solvable by ‘pure’ constraint propagation and few subpowers algorithms are fully characterized, see above, and fall way short of the CSPs that are conjectured to be polynomial time solvable according to the dichotomy conjecture. Designing ‘hybrid’ algorithms has therefore been the crucial problem in resolving the conjecture. Apart from the ad-hoc hybrid algorithms used for CSPs on small domains [Bulatov 2006b; Marković 2011; Zhuk 2016b; 2016a] and conservative CSPs [Bulatov 2011; 2003; Barto 2011; Bulatov 2016a], Maroti was the first who explicitly posed the problem of combining different algorithmic techniques for the CSP. Recall that a **congruence** of an algebra \( \mathbb{A} \) is an equivalence relation \( \theta \) invariant with respect to the operations of \( \mathbb{A} \). This way one may consider the quotient algebra \( \mathbb{A}/\theta \), whose elements are the equivalence classes of \( \theta \). Maroti attempted to prove the tractability of \( \text{CSP}(\mathbb{A}) \), in which algebra \( \mathbb{A} \) has a congruence \( \theta \) such that \( \mathbb{A}/\theta \) is an algebra with a Mal’tsev term, and every block of \( \theta \) satisfies the condition of Theorem 5.1, that is, gives rise to a CSP of bounded width; or the other way round \( \mathbb{A}/\theta \) has bounded width, while every \( \theta \)-block is Mal’tsev. He managed to design a hybrid algorithm for the former case [Maróti 2011a] and to make significant progress towards resolving the latter case [Maróti 2011b]. This latter case however turned out to be the crux in proving the dichotomy conjecture [Bulatov 2017a].

**Dichotomy theorems.** The dichotomy conjecture was settled independently and almost at the same time by the author [Bulatov 2017c; 2017b] (the Dichotomy Conjecture+++), and Zhuk [Zhuk 2017a; 2017b] (Conjecture 4.9) and [Zhuk 2018] (the Dichotomy Conjecture+++). The first algorithm, [Bulatov 2017c; 2017b] is based on the local structure of algebras as introduced in [Bulatov 2004; 2016b; 2016c] and a new notion of minimality. The second algorithm uses a totally different approach which involves an intricate combination of constraint propagation techniques and then approximating a solution using systems of linear equations. In this paper we give an outline of the first algorithm [Bulatov 2017c; 2017b].

**6. THE ALGORITHM**

We now outline the algorithm resolving the Dichotomy Conjecture. The main approach will be to introduce a more general minimality notion (not local anymore) that allows us to solve problems beyond bounded width, and then to reduce the general CSP to such instances. A more detailed description along with some simple examples can be found in [Bulatov 2017b; 2018].

**6.1. Algorithm ingredients**

**Gaussian Elimination and Few Subpowers.** The main routine of the algorithm is removing semilattice edges. Let \( \mathbb{A} = (A, F) \) be an idempotent algebra. A pair of elements...
$a, b \in \mathbb{A}$ is said to be a semilattice edge if there is a binary term operation $f$ of $\mathbb{A}$ such that $f(a, a) = a$ and $f(a, b) = f(b, a) = f(b, b) = b$, that is, $f$ is a semilattice operation on $\{a, b\}$. We say that algebra $\mathbb{A}$ is semilattice free if it does not contain semilattice edges. Removing semilattice edges is useful because of the following

**Proposition 6.1 ([Bulatov 2016c]).** If an idempotent algebra $\mathbb{A}$ is semilattice free, then it has few subpowers, and therefore $\text{CSP}(\mathbb{A})$ is solvable in polynomial time.

**Quasi-Centralizers.** Quasi-centralizer is an operator on the congruences of an algebra. It is similar to the centralizer as it is defined in commutator theory [Freese and McKenzie 1987], albeit the exact relationship between the two concepts is not quite clear, and so we name it differently for safety.

The set of all congruences of an algebra $\mathbb{A}$ is denoted by $\text{Con}(\mathbb{A})$. For an algebra $\mathbb{A}$, a term operation $f(x, y_1, \ldots, y_k)$, and $a \in \mathbb{A}^k$, let $f^a(x) = f(x, a)$; it is a unary polynomial of $\mathbb{A}$. Let $\alpha, \beta \in \text{Con}(\mathbb{A})$, and let $\zeta(\alpha, \beta) \subseteq \mathbb{A}^2$ denote the following binary relation: $(a, b) \in \zeta(\alpha, \beta)$ if and only if, for any term operation $f(x, y_1, \ldots, y_k)$, any $i \in [k]$, and any $a, b \in \mathbb{A}^k$ such that $a[i] = a, b[i] = b$, and $a[j] = b[j]$ for $j \neq i$, it holds $f^a(i) \subseteq \alpha$ if and only if $f^b(i) \subseteq \alpha$. (Polynomials of the form $f^a, f^b$ are sometimes called twin polynomials.) The relation $\zeta(\alpha, \beta)$ is always a congruence of $\mathbb{A}$.

**Proposition and algebras.** Let $\mathbb{A} = (A, F)$ be an algebra. As is mentioned in Section 3.2 applying propagation and other algorithms to an instance $I$ of $\text{CSP}(\mathbb{A})$ may change the domain of the variables of $I$. However, it is well known that these new domains remain subalgebras of $\mathbb{A}$, that is, operations from $F$ act on the new domains as well. The domain of a variable $v$ will be denoted $\mathbb{A}_v$. Transformations used in our algorithm can also change domain $\mathbb{A}_v$ to a quotient algebra $\mathbb{A}_{v/0}$. Thus, we will consider constraint relations as subsets of the product of different algebras: $R \subseteq \mathbb{A}_{v_1} \times \cdots \times \mathbb{A}_{v_k}$; however, the operations of $\mathbb{A}$ can still be used on these algebras.

**Decomposition of CSPs.** Let $R$ be a binary relation, a subset of the product of $\mathbb{A} \times \mathbb{B}$, and $\alpha, \gamma \in \text{Con}(\mathbb{A})$, and $\mathbb{A}, \mathbb{B}$ be algebras. Relation $R$ is said to be $\alpha, \gamma$-aligned if, for any $(a, c), (b, d) \in R$, $(a, b) \in \alpha$ if and only if $(c, d) \in \gamma$. This means that if $A_1, \ldots, A_k$ are the $\alpha$-blocks of $\mathbb{A}$, then there are also $k \gamma$-blocks of $\mathbb{B}$ and they can be labeled $B_1, \ldots, B_k$ in such a way that $R = (R \cap (A_1 \times B_1)) \cup \cdots \cup (R \cap (A_k \times B_k))$.

Let $I = (V, \mathbb{C})$ be a (2,3)-minimal instance from $\text{CSP}(\mathbb{A})$. We will always assume that a (2,3)-minimal instance has a constraint $C^X = (X, R^X)$ for every $X \subseteq V$, $|X| = 2$, where $R^X = S_X$. Recall that $\mathbb{A}_v$ denotes the domain of $v \in V$. A set $W \subseteq V$ is said to be a strand if it is maximal (under inclusion) among the sets with the following property: There are $\alpha_v \in \text{Con}(\mathbb{A}_v)$, $v \in W$, such that $R^{(v, w)}$ is $\alpha_v, \alpha_w$-aligned. Thus for a strand $W$ there is a one-to-one correspondence between $\alpha_v$- and $\alpha_w$-blocks of $\mathbb{A}_v$ and $\mathbb{A}_w$, $v, w \in W$. Moreover, by (2,3)-minimality these correspondences are consistent, that is, if $u, v, w \in W$ and $B_u, B_v, B_w$ are $\alpha_u$, $\alpha_v$, and $\alpha_w$-blocks, respectively, such that $R^{(u, v)} \cap (B_u \times B_v) \neq \emptyset$ and $R^{(v, w)} \cap (B_v \times B_w) \neq \emptyset$, then $R^{(u, w)} \cap (B_u \times B_w) \neq \emptyset$. This means that $I_W$ can be split into several instances, whose domains are $\alpha_v$-blocks.

**Lemma 6.2.** Let $I, W, \alpha_v$ for each $v \in W$, be as above. Then $I_W$ can be decomposed into a collection of instances $I_{1}, \ldots, I_{k}, k$ constant, $I_i = (W, C_i)$ such that every solution of $I_W$ is a solution of one of the $I_i$ and for every $v \in W$ its domain in $I_i$ is an $\alpha_v$-block.

**Subdirectly irreducible algebras.** In order to formulate the algorithm properly we need one more transformation of algebras. An algebra $\mathbb{A}$ is said to be subdirectly irreducible if the intersection of all its nontrivial (different from the equality relation)
denote the set of variables. Different methods of solving or reducing the instance to a strictly smaller one. In the algorithm we distinguish three cases depending on the presence of semilattice 2.3. The Algorithm

The notion of alignment allows for a new type of minimality of a CSP instance, block-minimality, which is key for our algorithm. In a certain sense it is similar to the standard local minimality, as it is also defined through a family of relations that have to be consistent in a certain way. However, block-minimality is not local, and is more difficult to establish, as it involves solving smaller CSP instances recursively. The definitions below are designed to allow for an efficient procedure to establish block-minimality.

Let \( I = (V, C) \) be a (2,3)-minimal instance. Then by solving a linear number of strictly smaller instances, we can reduce the problem \( I/\pi^v \) to one of \( I'/\pi_v \). It is also not hard to see that solving a linear number of strictly smaller CSPs, \( I \) can be transformed to an equivalent block-minimal instance \( I' \).

6.3. The Algorithm

In the algorithm we distinguish three cases depending on the presence of semilattice edges and quasi-centralizers of the domains of variables. In each case we employ different methods of solving or reducing the instance to a strictly smaller one. Let \( I = (V, C) \) be a subdirectly irreducible, (2,3)-minimal instance. Let \( \text{Center}(I) \) denote the set of variables \( v \in V \) such that \( \zeta(\Delta_v, \mu_v) \) is the full relation. Let \( \mu_v = \mu_v \) if \( v \in \text{MAX}(I) \cap \text{Center}(I) \) and \( \mu_v = \Delta_v \) otherwise.
Semilattice Free Domains. If no domain of \( I \) contains a semilattice edge then by Proposition 6.1 \( I \) can be solved in polynomial time, using the few subpowers algorithm, as shown in [Idziak et al. 2010; Bulatov 2016c].

Small Centralizers. If \( \mu_v^* = \Delta_v \) for all \( v \in V \), block-minimality guarantees the existence of a solution, as Theorem 6.4 shows, and we can use Lemma 6.3 to solve the instance.

**Theorem 6.4.** If \( I \) is subdirectly irreducible, \((2,3)\)-minimal, block-minimal, and \( \text{MAX}(I) \cap \text{Center}(I) = \emptyset \), then \( I \) has a solution.

Large Centralizers. Suppose that \( \text{MAX}(I) \cap \text{Center}(I) \neq \emptyset \). In this case the algorithm proceeds in three steps.

**Step 1.** Consider the problem \( I/\mu^* \). We establish the global 1-minimality of this problem. If it is changed in the process, we start solving the new problem from scratch. Checking global 1-minimality can be reduced using standard techniques to solving a linear number of problems that are either strictly smaller, or have small centralizers (see above).

**Step 2.** For every \( v \in \text{Center}(I) \) we find a solution \( \varphi \) of \( I/\mu^* \) satisfying the following condition: there is \( a \in \mathbb{A}_v \) such that \( \{a, \varphi(v)\} \) is a semilattice edge if \( \mu_v^* = \Delta_v \), or, if \( \mu_v^* = \mu_v \), there is \( b \in \varphi(v) \) such that \( \{a, b\} \) is a semilattice edge. Such a solution exists since \( I/\mu^* \) is globally 1-minimal.

**Step 3.** Using the solutions found in Step 2 apply the transformation of \( I \) suggested by Maroti in [Maróti 2011b]. This transformation results in an equivalent instance, but eliminates from the respective domains the ‘lower’ ends \( a \) of semilattice edges chosen in Step 2. Thus the resulting instance is strictly smaller.

Using Lemma 6.3 and Theorems 6.4 it is not difficult to see that the algorithm runs in polynomial time. Indeed, every time it makes a recursive call it calls on a problem whose non-semilattice free domains of maximal cardinality have strictly smaller size, and therefore the depth of recursion is bounded by \( |\mathbb{A}| \) if we are dealing with \( \text{CSP}(\mathbb{A}) \). More precisely,

**Theorem 6.5.** If \( \mathbb{A} \) is a finite idempotent algebra with a weak near-unanimity term. Then \( \text{CSP}(\mathbb{A}) \) can be solved in time \( O(Nmn|\mathbb{A}|^k) \), where \( n \) is the number of variables in an instance, \( m \) the number of constraints, \( N \) is the total number of tuples in all the constraint relations, and \( k \) is a constant such that the CSP over algebras with few subpowers derived from \( \mathbb{A} \) can be solved in time \( O(mn^k) \).

7. FUTURE DIRECTIONS

We conclude the column with a short review of open questions related to the dichotomy conjecture, related areas and potential future directions.

Polymorphism oblivious algorithms. There is a peculiar asymmetry between the two main types of CSP algorithms, constraint propagation and the few subpowers algorithm. While constraint propagation can be run on any given instance without any prior knowledge about the underlying constraint language or algebra (although also without any guarantees to solve the problem), the few subpowers algorithm explicitly uses the polymorphisms associated with the problem. Both general algorithms for the CSP also use the knowledge of the algebraic structure of the problem. It is therefore an important question whether or not there exists an algorithm that solves, say, few subpowers CSPs without knowing any polymorphisms of the constraint language, but only certain local properties of the relations involved.
This question has a connection to the problem of recognizing, given a relational structure \( \mathcal{H} \) or an algebra \( \mathcal{A} \), if the problem \( \text{CSP}(\mathcal{H}) \) or \( \text{CSP}(\mathcal{A}) \) can be solved in polynomial time or has bounded width, or is within some other complexity class. This problem is known as the metaproblem, see [Chen and Larose 2017; Freese and Valeriote 2009]. Chen and Larose in [Chen and Larose 2017] observed that if a class of CSPs has such a polymorphism oblivious algorithm, then the metaproblem for this class can be solved in polynomial time (assuming the structures involved are cores and algebras are idempotent). In particular, the metaproblem for the class of structures of bounded width is polynomial time, while for the class of structures with tractable CSP the complexity of the metaproblem is unknown.

Other complexity classes. There is strong evidence that nonuniform CSPs can be complete in very few complexity classes. In [Allender et al. 2009] Allender et al. showed that for constraint language \( \Gamma \) on a 2-element set, the problem \( \text{CSP}(\Gamma) \) can be complete in only a handful of complexity classes: NP, P, \( \oplus P \), NL, L, and \( AC^0 \). A similar classification has been conjectured in the general case by Larose and Tesson [Larose and Tesson 2009] with the class \( \text{mod}_p L \) for prime \( p \) instead of \( \oplus P \). The best way to express this collection of conjectures is through omitting types of the local structure of algebras in the sense of tame congruence theory [Hobby and McKenzie 1988]. Assuming all the complexity classes involved are different, for a structure \( \mathcal{H} \) the problem \( \text{CSP}(\mathcal{H}) \)

(a) is NP-complete unless \( \text{Alg}(\mathcal{H}) \) omits the unary type,
(b) is \( \text{mod}_m L \)-complete for some \( m \) if and only if \( \text{Alg}(\mathcal{H}) \) omits the unary and semilattice types but does not omit the affine type ([Larose and Tesson 2009] shows that in this case \( \text{CSP}(\mathcal{H}) \) is \( \text{mod}_p L \)-hard for some prime \( p \)),
(c) is P-complete if and only if \( \text{Alg}(\mathcal{H}) \) omits the unary type but not the semilattice type,
(d) is of bounded width if and only if \( \text{Alg}(\mathcal{H}) \) omits the unary and affine types,
(e) is NL-complete if and only if \( \text{Alg}(\mathcal{H}) \) omits the unary, affine and semilattice types but not the lattice type,
(f) is L-complete if and only if \( \text{Alg}(\mathcal{H}) \) omits the unary, affine, semilattice, and lattice types, but \( \text{CSP}(\mathcal{H}) \) is not FO-expressible.

The hardness parts of all these conjectures are confirmed in [Larose and Tesson 2009]. Items (a) and (d) are the dichotomy theorem and the characterization of CSPs of bounded width which are also established. FO-expressible problems have been characterized in [Larose et al. 2007]. Kazda [Kazda 2018] proved that (e) implies (f). Finally, Dalmau and Krokhin [Dalmau and Krokhin 2008], and Barto et al. [Barto et al. 2012] made significant progress towards resolving (e). The rest of the problems above remain wide open.

Infinite CSPs. The majority of work on the CSP has been done under the assumption that the domain is finite. Allowing infinite domains expands the CSP framework so that it includes an enormous range of problems from GRAPH-SAT [Bodirsky and Pinsker 2015] to problems of scheduling and temporal reasoning [Allen 1983; Jonsson and Krokhin 2004; Bodirsky et al. 2018]. Problems representable by infinite CSPs such as temporal and spatial reasoning are standard in artificial intelligence. However, there has also been a significant amount of research initiated by [Bodirsky and Nešetril 2003] on the algebraic structure of such problems. Although infinite CSPs use a variety of specific methods, the overall approach is to identify a finite algebraic structure in an as large as possible class of infinite CSPs [Barto and Pinsker 2016; Barto et al. 2017; Pinsker 2015]. The current dichotomy conjecture for infinite CSPs [Barto and Pinsker 2016; Barto et al. 2017] extends that for finite CSPs. For a recent survey on infinite CSPs see [Bodirsky and Mamino 2017].
Alternative parametrizations. In nonuniform CSPs we restrict a constraint language or a template relational structure. Clearly, other kinds of restrictions are also possible. For instance, in database theory one cannot assume any restrictions on the possible content of a database — which is a template structure in the CONJUNCTIVE QUERY EVALUATION problem — but some restrictions on the possible form of queries make much sense. If a CSP is viewed as in Definition 2.1, the constraint scopes of an instance $I$ form a hypergraph on the set of variables. In a series of works [Gottlob et al. 2000; Flum et al. 2002; Gottlob et al. 2002; Grohe 2007; Grohe and Marx 2014] it has been shown that if this hypergraph allows some sort of decomposition, or is tree-like, then the CSP can be solved in polynomial time. The tree-likeness of a hypergraph is usually formalized as having bounded tree width, or bounded hypertree width, or bounded fractional hypertree width. This line of work culminated in [Marx 2013], in which Marx gave an almost tight description of classes of hypergraphs that give rise to a CSP solvable in polynomial time. Hybrid restrictions are also possible, although research in this direction has been more limited, see, [Feder et al. 2003; Feder and Hell 2006; Cooper and Zivny 2017] as an example.

The Promise CSP. Recently, Brakensiek and Guruswami [Brakensiek and Guruswami 2018b] suggested the following generalization of the CSP that they called the Promise CSP. An instance of the PCSP consists of a pair of CSP instances $(I, I')$ such that they have the same number of constraints and for each constraint $(s, R)$ of $I$ there is a constraint $(s, R')$ of $I'$ such that $R \subseteq R'$. The goal is to distinguish between the case when $I$ is satisfiable and the case when $I'$ is unsatisfiable. PCSP can express a much wider class of problems than the regular CSP, which includes, for instance, approximate graph and hypergraph colouring. It also uses a wider variety of solution algorithms such as LP and combinations of LP with other techniques [Brakensiek and Guruswami 2018b; 2018a]. On the other hand, PCSP allows for algebraic approach (although more limited than the regular CSP) as was demonstrated in [Brakensiek and Guruswami 2018b] and further developed in [Krokhin and Opršal 2018]. In the latter work a connection between the LABEL COVER problem and checking the triviality of certain systems of algebraic identities has been established.

Variations of the CSP. Numerous variations and generalizations of the regular CSP have been studied over the last two decades. These include quantified CSPs, counting CSP, enumeration problems, CSPs with global constraints, a number of optimization problems such as Max- and Min-CSP, Valued CSP, the Min-Homomorphism problem. Many of the counting and optimization problems admit approximation algorithms, which have also been extensively studied. A dichotomy or other complexity classification results have been proved (or sometimes conjectured) for a number of those problems starting from the early works for 2-element structures, see, [Creignou et al. 2001]. A dichotomy theorem has been proved for the counting CSP [Bulatov 2013; Cai and Chen 2017]. Similarly, for the optimization problem of (Valued) CSP a dichotomy result is proved in [Thapper and Zivny 2016; Kolmogorov et al. 2017], and a (conditional) complexity classification of approximation of Valued CSP [Raghavendra 2008] was established. We should also mention the recent advances in complexity classification of Quantified CSP, enumeration problems, and a large number of related problems, in each of which the hope is to obtain some dichotomy-like results. Unfortunately, there is no room in this column to stop even briefly on any of these fascinating problems; each of them requires its own survey. The keen reader is however referred to a recent collection of such surveys [Krokhin and Zivny 2017].

REFERENCES


In this accessible yet rigorous column, Christel Baier and Clemens Dubslaff provide a timely and valuable overview of the vibrant area of algorithmic synthesis problems based on Markov decision processes. The latter are one of the most prominent stochastic models, both in theory and in applications. The overview is extensive, covering synthesis with non-standard objectives as well as family-based analysis, and indicating several emerging directions.
Various algorithms and tools for the formal verification of systems with respect to their quantitative behavior have been developed in the past decades. Many of these techniques inherently support the automated synthesis of strategies that guarantee the satisfaction of performance or reliability constraints by resolving controllable nondeterministic choices in an adequate way. More recently, such techniques have been further developed towards the analysis or strategy synthesis under multiple cost and utility constraints.

This article deals with Markov decision processes (MDPs) for modeling systems and their environment and provides an overview of recent directions for different types of synthesis problems in MDPs that aim to achieve an acceptable cost-utility tradeoff.

1. INTRODUCTION

Markov decision processes (MDPs, see, e.g., [Puterman 1994; Filar and Vrieze 1996]), are a prominent stochastic model that has been introduced in the 1950s and widely used for various types of optimization problems with applications, e.g., in operations research, reinforcement learning, and robotics. Since the 1990s, MDPs have been used as an operational model for distributed probabilistic systems and various verification algorithms for MDPs and temporal logics have been developed. In this setting, the nondeterministic choices are typically viewed as choices made by an adversary and the classical verification task is to show that a threshold condition for the probability of a temporal path property or an expectation constraint holds, no matter how the nondeterministic choices are resolved. For many types of temporal properties or expectations, worst-case strategies for resolving the nondeterministic choices exist in the sense that they maximize or minimize the probabilities resp. expectations under consideration. Indeed, many verification algorithms inherently compute such extremal strategies, which turns the classical verification problem for MDPs into a strategy-synthesis problem. In the past decade, there has been a drift from classical “single-objective” verification or synthesis problems to objectives that combine cost and utility constraints, defined, e.g., through weight functions, temporal logics, or combinations thereof. As a matter of fact, even the traditional shortest-path problem for MDPs [Bertsekas and Tsitsiklis 1991] is inherently multi-objective by combining a qualitative utility condition “reaching the target almost surely” with the cost objective “minimize the expected cost.”

---

1The authors are supported by the DFG through the collaborative research centre HAEC (SFB 912) and projects BA 1679/11-1 and 1679/12-1.
total cost”. In the past, several variants of shortest-path problems and other analysis and synthesis problems under tradeoff constraints to balance multiple cost and utility objectives have been considered. Following [Chatterjee et al. 2015], quantitative measures for how well does a system satisfy a specification can be classified as follows:

1. measures along behaviors, such as the cost or reward until reaching a target or the long-run average cost or reward (e.g., mean payoff, long-run cost-utility ratios),
2. measures across behaviors, such as worst- or average-case behaviors, but also quantiles or conditional probabilities and expectations.

The development of algorithms, complexity-theoretic considerations as well as the investigation of resource requirement for worst- or best-case strategies according to specifications combining measures of type (1) and/or (2) are very active research fields.

In this article, we provide a summary of classical results and a selection of more advanced recent developments in these directions. We start in Section 2 with some basic concepts and recent achievements on how to solve strategy-synthesis problems. Other types of synthesis problems for Markovian models will be addressed in Sections 3 and 4. While Section 3 gives an overview on variant-selection problems using feature-oriented formalisms, Section 4 provides insights in the potential of parameter-synthesis approaches for Markovian models with parametric transition probabilities.

2. WORST-CASE ANALYSIS AND STRATEGY SYNTHESIS

To explain the main ideas of synthesis approaches, we briefly introduce some basic notations for Markov decision processes (MDPs) where we restrict ourselves to MDPs over a finite state space and action set. Details can be found in textbooks such as [Puterman 1994; Filar and Vrieze 1996] or in [Baier and Katoen 2008; Baier et al. 2018] for details on the use of MDPs as an operational system model and algorithms for the formal analysis against temporal-logic specifications. We use standard linear temporal logic (LTL) like notations to denote path properties, e.g., $\Diamond G$ for the reachability condition “eventually visit a $G$-state” or $\square \Diamond G$ for the Büchi condition “infinitely often $G$” or $\square \Diamond G$ for the co-Büchi condition “almost always $G$”.

Given a finite set $S$, a distribution on $S$ is a function $\mu : S \rightarrow [0, 1]$ with $\sum_{s \in S} \mu(s) = 1$. We write $\text{Distr}(S)$ for the set of distributions $\mu$ on $S$ where $\mu(s) \in [0, 1]$ for all $s \in S$. A (plain) MDP is a tuple $M = (S, Act, P)$ where $S$ is a finite set of states, $Act$ a finite set of actions, and $P : S \times Act \rightarrow \text{Distr}(S)$ a partial probability function.

For $s \in S$ we denote by $\text{Act}(s) = \{ \alpha \in Act : P(s, \alpha) \neq \bot \}$ the set of actions that are enabled in state $s$, where $\bot$ indicates “undefined”. State $s$ is called a trap if $\text{Act}(s) = \emptyset$. The special case of a (finite-state) Markov chain is obtained when $Act$ is a singleton, in which case the action name is irrelevant and $P$ can be treated as a partial function $P : S \times S \rightarrow [0, 1] \cap \mathbb{Q}$.

An end component of $M$ is a strongly connected sub-MDP. They can be specified by sets of state-action pairs. While the number of end components can be exponential, the number of maximal end components (i.e., end components that are not contained in larger end components) is bounded by $|S|$. Maximal end components are computable in polynomial time [de Alfaro 1997; Chatterjee and Henzinger 2011].

A finite path in $M$ is a sequence $\pi = s_0 a_0 s_1 a_1 \ldots a_{n-1}s_n$ where $s_0, s_1, \ldots, s_n \in S$ are states and the $a_i$’s are actions with $a_i \in \text{Act}(s_i)$ and $P(s_i, a_i, s_{i+1}) > 0$ for all $i < n$. We write first($\pi$) resp. last($\pi$) for the first resp. last state of $\pi$, i.e., if $\pi$ is as above then first($\pi$) = $s_0$ and last($\pi$) = $s_n$. Infinite paths are defined accordingly. A path is said to be maximal if it is either infinite or finite and ends in a trap.

A strategy (a.k.a. scheduler, policy, adversary) for $M$ is a function that assigns to each non-maximal finite path $\pi$ a distribution $\mathcal{S}(\pi) \in \text{Distr}(\text{Act(last(\pi)))}$. $\mathcal{S}$ is called deter-
ministic if for each non-maximal path $\pi$ there is a unique action $\alpha$ with $G(\pi)(\alpha) = 1$, and memoryless if $G$'s decision only depends on the last state of the input path. Finite-memory strategies are those strategies that can be realized by a finite-state machine.

Given a strategy $G$ and a state $s \in S$, we write $P_{\sup G}^{\sup G}$ or briefly $P_{\sup}^{\sup G}$ for the probability measure on (measurable) sets of maximal paths starting in $s$ induced by the Markov chains obtained by unfolding $M$ from $s$ into a tree following $G$'s decisions. Note that the Markov chains of finite-memory strategies have a regular structure and can be collapsed into finite-state Markov chains. If $\varphi$ is a measurable path property then we simply write $P_{\sup G}^{\sup G}(\varphi)$ for the probability measure of all $G$-paths from $s$ satisfying $\varphi$. For a worst- or best-case analysis one ranges over all strategies (i.e., all possible resolutions of the nondeterminism) and considers the extremal probabilities for satisfying $\varphi$:

$$P_{\sup M,s}^{\sup}(\varphi) = \sup_{G} P_{\sup G}^{\sup G}(\varphi) \quad \text{and} \quad P_{\inf M,s}^{\inf}(\varphi) = \inf_{G} P_{\inf G}^{\inf G}(\varphi).$$

In case the supremum resp. infimum is met, we write $P_{\sup M,s}^{\max}(\varphi)$ resp. $P_{\inf M,s}^{\min}(\varphi)$. This, e.g., applies if $\varphi$ is an $\omega$-regular property, in which case even optimal finite-memory strategies exist.

Plain MDPs can be extended by several additional features such as a declaration of the initial states, state labels for temporal-logic specifications or weight functions for reasoning about cost and reward constraints. A weighted MDP is an MDP with a weight function $\mathrm{wgt} : S \times \text{Act} \to \mathbb{Q}$ that assigns rational weights to all state-action pairs. The accumulated weight of a finite path $\pi$ is denoted by $\mathrm{wgt}(\pi)$ and is defined as the sum of weights of its state-action pairs, i.e.,

$$\mathrm{wgt}(s_0\alpha_0 s_1\alpha_1 \ldots s_n\alpha_n) = \mathrm{wgt}(s_0, \alpha_0) + \mathrm{wgt}(s_1, \alpha_1) + \ldots + \mathrm{wgt}(s_n, \alpha_n).$$

Given an infinite path $\varsigma = s_0\alpha_0 s_1\alpha_1 \ldots$, the upper resp. lower mean payoff is defined by

$$\overline{\mathrm{MP}}(\varsigma) = \lim_{n \to \infty} \frac{1}{n} \mathrm{wgt}(s_0\alpha_0 \ldots s_{n-1}s_n) \quad \text{and} \quad \underline{\mathrm{MP}}(\varsigma) = \lim_{n \to \infty} \frac{1}{n} \mathrm{wgt}(s_0\alpha_0 \ldots s_{n-1}s_n).$$

It is well known that if $M$ is a strongly connected Markov chain, then

$$\overline{\mathrm{MP}}(\varsigma) = \underline{\mathrm{MP}}(\varsigma) = \sum_{s \in S} \theta(s) \cdot \mathrm{wgt}(s)$$

for almost all infinite paths $\varsigma$. Here, $\theta(s)$ denotes the long-run frequency of $s$ and $\mathrm{wgt}(s)$ denotes the weight of the unique state-action pair $(s, \alpha)$ in $M$. The minimal and maximal expected mean payoff in MDPs can be achieved by deterministic memoryless strategies, both computable in polynomial time using linear-programming (LP) techniques. The key idea for strongly connected MDPs is to use one variable $y_{s,\alpha}$ for the long-run frequencies of the state-action pairs $(s, \alpha)$ under optimal memoryless randomized strategies. This approach has been extended for general MDPs by using additional variables for the frequencies of the state-action pairs in the transient part [Kallenberg 1983]. Alternatively, one can first determine the maximal end components of the given MDP, compute the maximal or minimal expected mean payoff for them using LP techniques for strongly connected MDPs and finally compute the maximal or minimal expected mean payoff in $M$. This is done by computing the maximal resp. minimal expected accumulated weight in an MDP that results from $M$ by collapsing each maximal end component $E$ into a single state and adding a deterministic transition from $E$ to a fresh goal state whose weight is the extremal expected mean payoff in $E$. 

---

ACM SIGLOG News 28 October 2018, Vol. 5, No. 4
2.1. Bellman equations and the classical stochastic shortest-path problem

The stochastic shortest-path problem (SSP) is an optimization problem that aims to find a policy minimizing the expected costs until reaching a target. In this context, we consider a weighted MDP $\mathcal{M} = (S, \text{Act}, P, \text{wgt})$ with a distinguished target state $\text{goal} \in S$ such that $P_{\text{goal}}^{s_{max}}(\diamond \text{goal}) = 1$ for all states $s \in S$. The task of the SSP is to determine a proper strategy that minimizes the expected accumulated weight until reaching the goal state from all states $s$ where the minimum is taken over all proper strategies. Here, a strategy $\mathcal{S}$ is called proper in case $P_{\mathcal{S},s}^{s_{goal}}(\diamond \text{goal}) = 1$ for all states $s$.

**Theorem 2.1 ([Bertsekas and Tsitsiklis 1991; de Alfaro 1999; Baier et al. 2018]).**

SSP is solvable in polynomial time.

Let us briefly sketch how to solve the SSP in polynomial time. Given a proper strategy $\mathcal{S}$ and state $s \in S$, let $E_{\mathcal{S},s}^{\text{goal}}(\diamond)$ denote the expectation of the random variable $\diamond$ that assigns to each path ending in state $s$ its accumulated weight. Let $x_s = E_{\mathcal{S},s}^{\text{goal}}(\diamond)$ denote the infimum over all proper strategies. For simplicity, let us suppose that $\diamond$ is a trap that is accessible from all states which ensures the existence of proper strategies. Then, the vector $(x_s)_{s \in S}$ satisfies the Bellman equations given by

$$x_s = \min \left\{ \text{wgt}(s, \alpha) + \sum_{t \in S} P(s, \alpha, t) \cdot x_t : \alpha \in \text{Act}(s) \right\}$$

for all states $s \in S \setminus \{\text{goal}\}$. The classical setting of [Bertsekas and Tsitsiklis 1991] assumes that for each improper strategy $\mathcal{S}$ there is at least one state $s \in S$ such that the expected total weight from $s$ under $\mathcal{S}$ is $+\infty$. We refer to the latter as the SSP-assumption. This assumption ensures that the fixed-point operator given by the Bellman equations is a contracting map in a Banach space and thus has a unique solution, obtainable as the greatest solution of the following linear constraints: $x_{\text{goal}} = 0$ and

$$x_s \leq \text{wgt}(s, \alpha) + \sum_{t \in S} P(s, \alpha, t) \cdot x_t \quad \text{for all } s \in S \setminus \{\text{goal}\} \text{ and } \alpha \in \text{Act}(s).$$

This yields the polynomial-time solvability for MDPs satisfying the SSP-assumption. In [de Alfaro 1999], the cases of MDPs where either all weights are non-negative or all weights are non-positive have been considered. Furthermore, [de Alfaro 1999] presented polynomial-time algorithms to check the finiteness of the values $x_s$ in such an MDP $\mathcal{M}$, and if so, to transform $\mathcal{M}$ into an equivalent MDP $\mathcal{M}'$ satisfying the SSP-assumption. More recently, [Baier et al. 2018] shows how to solve the SSP for the general case. Speaking roughly, while [de Alfaro 1999] ensures the SSP-assumption in MDPs with non-positive weights by collapsing end components consisting of state-action pairs $(s, \alpha)$ with $\text{wgt}(s, \alpha) = 0$ into a single state, [Baier et al. 2018] generalizes this approach by successively flattening end components where the accumulated weight of all cycles is 0 (called 0-ECs) using the so-called spider construction. The idea of the latter is to exploit the fact that for each pair $(s, t)$ of states in a 0-EC $\mathcal{E}$ there exists a value $w(s, t) \in \mathbb{Q}$ such that the accumulated weight of all paths from $s$ to $t$ inside $\mathcal{E}$ have weight $w(s, t)$. In this case we can simply pick an arbitrary reference state $s_0$ in $\mathcal{E}$, discard all state-action pairs $(s, \alpha) \in \mathcal{E}$ and introduce deterministic transitions from each state $s \neq s_0$ to $s_0$ with weight $w(s, s_0)$. Finally, we replace all state-action pairs

---

2 More precisely, [Baier et al. 2018] considers MDPs with integer weight functions. However, the approach of [Baier et al. 2018] is also applicable for rational weight functions by multiplying all weights with the least common multiple of the denominators of the weights.
(s, β) of M where s is a state of E different from s₀ and (s, β) \notin E with the new state-action pair (s₀, β*) where \text{wgt}(s₀, β*) = w(s₀, s) + \text{wgt}(s, β). This yields a new MDP with the same state space and the same minimal expected accumulated weight until reaching goal but fewer state-action pairs. The procedure can be repeated until either an end component E with negative minimal expected mean payoff has been found, in which case x_s = -\infty for all states in E, or the minimal expected mean payoff of all end components is positive, in which case the generated MDP satisfies the SSP-assumption.

### 2.2. Model checking MDPs against temporal-logic specifications

Probabilistic computation-tree logic (PCTL) [Hansson and Jonsson 1994; Bianco and de Alfaro 1995] has been introduced as a variant of CTL where the CTL path quantifiers “there is a path satisfying φ” and “all paths satisfy φ” are replaced with a probability operator. Formulas with the probability operator have the form \text{Pr}[φ] where φ is a path formula and I \subseteq [0, 1] an interval with rational end points. Path formulas are built as in CTL using the next and until operator as basic temporal modalities and PCTL state formulas as their arguments. When interpreting PCTL formulas over an MDP M with state labels, the semantics of \text{Pr}[φ] is the set of all states s in M such that \text{Pr}^s_{M,s}(φ) \in I for all strategies E. The task of the PCTL model-checking problem is to decide whether a given initial state of an MDP M satisfies a given PCTL state formula φ.

**Theorem 2.2** (Hansson and Jonsson 1994; Bianco and de Alfaro 1995). 

*The PCTL model-checking problem for MDPs (and Markov chains) is in P.*

The standard PCTL model-checking algorithm computes the satisfaction sets of all subformulas of \(\Phi\) in a bottom-up manner (“inner subformulas first”) and finally checks whether the distinguished initial state belongs to the satisfaction set of \(\Phi\). Let us consider the PCTL formula \(\text{Pr}[\Psi_1 \cup \Psi_2]\) where \(\Psi_1, \Psi_2\) are state formulas (whose satisfaction sets have been computed before and thus can be treated as sets of states) and \(\cup\) stands for the standard until operator. Then, the satisfaction set of \(\text{Pr}[\Psi_1 \cup \Psi_2]\) is obtained by a polynomial-time graph analysis to determine all states \(s\) where \(\text{Pr}^s_{M,s}(\Psi_1 \cup \Psi_2)\) equals 0 or 1 and then computing the maximal resp. minimal probabilities for the path property \(\Psi_1 \cup \Psi_2\) when ranging over all strategies. Indeed, memoryless strategies maximizing or minimizing the probabilities \(\Psi_1 \cup \Psi_2\) from all states exist and can be computed by linear-programming (LP) techniques. The encoding of maximal or minimal probabilities for reachability (resp. until) properties is similar to the Bellman equations for minimal expected accumulated weights until reaching a target (see Section 2.1).

For more complex path properties specified, e.g., using linear temporal logic (LTL), the typical task of the analysis is to check whether the probability for a given initial state and a given LTL formula \(φ\) meets a threshold condition under all strategies. This corresponds to a worst- or best-case analysis depending on whether \(φ\) is a desired or undesired event.

**Theorem 2.3** (Vardi 1985; Vardi and Wolper 1986; Courcoubetis and Yannakakis 1995). 

*The LTL model-checking problem in MDPs is 2EXPtime-complete, even in the qualitative case where the task is to decide whether an LTL formula holds almost surely or with positive probability under each or some strategy.*

The standard methods to solve the LTL model-checking problem for an MDP M solve the induced synthesis problem that asks to determine a strategy maximizing or minimizing the probability for \(φ\) from a given initial state. Such strategies can be obtained by first generating a deterministic automaton A for \(φ\), constructing a product-
MDP $\mathcal{M} \times \mathcal{A}$ and then computing a memoryless strategy for $\mathcal{M} \times \mathcal{A}$ that maximizes or minimizes the probability for reaching an end component that meets $\mathcal{A}$'s acceptance condition. As the translation of LTL formulas into deterministic automata can cause a doubly exponential blow-up, this approach yields a double exponential upper bound for the LTL model-checking problem in MDPs.

More advanced techniques for end components and several types of long-run properties, including long-run cost-utility ratios, have been proposed in [de Alfaro 1997] and [von Essen et al. 2016].

2.3. Other strategy-synthesis problems
Besides refinements of strategy-synthesis algorithms with SSP-style expectation objectives or temporal-logic specifications, several other strategy-synthesis problems have been addressed in the past decade. We provide here a brief summary of a selection of such research directions.

2.3.1. Multi-objective strategy synthesis. While the classical synthesis problems for MDPs ask to find a strategy that minimizes or maximizes the probability for a temporal property or the expectation of accumulated or discounted costs, the multi-objective synthesis problems aim to find a strategy that satisfies multiple probability and/or expectation constraints, possibly with some optimization criterion under all those strategies. For example, a typical instance of a multi-objective synthesis problem might ask to find a strategy that maximizes the expected utility subject to the requirement that the consumed energy stays below a given threshold with probability at least 0.99.

Pareto realizability for MDPs with multiple discounted reward objectives has been studied in [Chatterjee et al. 2006]. These results have been generalized in [Chatterjee et al. 2013b] for a more expressive class of models that allow to specify discount factors depending on states and/or objectives.

In [Etessami et al. 2008], a linear-programming (LP) approach to solve the synthesis problem for MDPs with multiple probabilistic reachability constraints has been proposed. The crucial observation is that randomized memoryless strategies are sufficient for this task and one can use LP techniques to check the existence of such strategies and to compute them using linear constraints with variables encoding the frequencies of the state-action pairs under optimal strategies. Analogous LP techniques have been proposed in [Forejt et al. 2011; Kwiatkowska et al. 2013] to treat combinations of constraints on expected accumulated costs and probability constraints for $\omega$-regular path properties. These techniques have been employed for the compositional assume-guarantee verification for MDPs. Algorithms for multi-objective cost-bounded reachability conditions in MDPs are provided in [Randour et al. 2015; Hartmanns et al. 2018].

MDPs with multiple mean-payoff objectives in terms of expectation or probability constraints have been considered in [Brázdil et al. 2014]. For multiple expectation constraints, [Brázdil et al. 2014] proposes polynomial-time algorithms to check feasibility and to compute optimal 2-memory stochastic-update strategies. The treatment of multiple probabilistic constraints is more difficult as optimal strategies might require infinite memory. However, $\varepsilon$-optimal strategies are shown to be computable in polynomial time.

The “beyond worst-case” synthesis approach of [Bruyère et al. 2014] addresses the problem to find a strategy that is optimal with respect to an expected behavior, while satisfying a path property along all its paths. More precisely, [Bruyère et al. 2014] considers the SSP problem in combination with strict guarantees for a mean-payoff condition and presents a pseudo-polynomial algorithm for it. The decision problem is
shown to be in $\text{NP} \cap \text{coNP}$ and at least as hard as (non-stochastic) two-player mean-payoff games.

2.3.2. Energy conditions. Energy MDPs are weighted MDPs where the integer weight function formalizes the change of an energy budget to model, e.g., the power level of a battery. Let $\mathcal{M}$ be an energy MDP with weight function $\text{wgt}: S \times \text{Act} \rightarrow \mathbb{Z}$. A path $\pi$ in $\mathcal{M}$ fulfills the energy objective w.r.t. a given initial energy budget $c \in \mathbb{N}$ when for all prefixes $\rho$ of $\pi$ we have $\text{wgt}(\rho) \geq 0$. Intuitively, an energy objective w.r.t. $c$ stands for the requirement that the system never runs out of energy provided by a battery of capacity $c$. The energy objective has been introduced for quantitative models as a safety-like condition that requires the accumulated weight (e.g., available energy) to stay continuously above a given threshold [Bouyer et al. 2008]. Using parity conditions encoding $\omega$-regular side constraints enables reasoning about important system properties under the energy objective:

**Theorem 2.4 ([Chatterjee and Doyen 2011; Mayr et al. 2017]).** The problem of deciding whether there exists an initial credit $e$ and a strategy that achieves the energy objective w.r.t. $e$ and a given parity condition is in $\text{NP} \cap \text{coNP}$.

Several related problems have been considered for energy MDPs. In [Brázdil et al. 2016], energy MDPs with an additional non-negative weight function (standing for the payoff achieved in some transition) have been considered, synthesizing strategies that ensure the energy condition and maximizing the expected mean payoff. Closely related to energy MDPs are one-counter MDPs with boundary [Brázdil et al. 2010; Brázdil et al. 2013; Etessami and Yannakakis 2015], which can be seen as weighted MDPs where all weights are in $\{-1, 0, +1\}$ and terminate as soon as the counter value is 0. The termination problem asks whether there is a strategy that guarantees almost-sure termination (i.e., reaching a configuration where control is in a final location and the counter has value 0). An exponential-time algorithm for deciding the termination problem in one-counter MDPs has been presented in [Brázdil et al. 2013], also showing $\text{PSPACE}$-hardness of this problem. In contrast, for weighted MDPs that do not have a lower bound on the counter values (accumulated weights), the analogous problem of almost-sure reaching a target along paths where the accumulated weight is bounded from below (or above) by a given constant under some strategy is in $\text{NP} \cap \text{coNP}$ and at least as hard as two-player mean-payoff games [Baier et al. 2018].

2.3.3. Quantiles and conditionals. A well-established concept for the synthesis of strategies in MDPs that optimize the cost-utility tradeoff relies on quantiles (see, e.g., [Ummels and Baier 2013; Baier et al. 2014a; Baier et al. 2014b]). Let $\mathcal{M} = (S, \text{Act}, P, \text{wgt})$ be a weighted MDP with non-negative weight function $\text{wgt}: S \times \text{Act} \rightarrow \mathbb{N}$, a state $s \in S$, $p \in [0, 1] \cap \mathbb{Q}$ a probability threshold, and $\gg \in \{\gg, \geq\}$ a binary relation. In the sequel, we refer to non-negative weights as costs. Then, quantiles w.r.t. probabilistic constraints for until properties on state sets $A, B \subseteq S$ with upper or lower cost bounds and lower probability bounds are provided through

universal quantiles:  
$\min \{ c \in \mathbb{N} : \text{Pr}_{M,s}^{\text{min}}(A U^{<c} B) \geq p \}$ \quad \min \{ c \in \mathbb{N} : \text{Pr}_{M,s}^{\text{max}}(A U^{\leq c} B) \geq p \}$
existential quantiles:  
$\max \{ c \in \mathbb{N} : \text{Pr}_{M,s}^{\text{min}}(A U^{\geq c} B) \geq p \}$ \quad \max \{ c \in \mathbb{N} : \text{Pr}_{M,s}^{\text{max}}(A U^{> c} B) \geq p \}$

For instance, the existential quantile $\min \{ c \in \mathbb{N} : \text{Pr}_{M,s}^{\text{max}}(A U^{\leq c} B) > p \}$ asks for the minimal cost $c$ that need to be spent to guarantee reaching $B$ through $A$-states with probability exceeding $p$. Encoding a second non-negative integer weight function into the state space of $\mathcal{M}$ and formalizing a notion of utility enables to reason about the cost-
utility tradeoff by only regarding states in $B$ that exceed a given utility threshold. While qualitative quantiles, i.e., $\geq p$ is either “$> 0$” or “$\leq 1$”, are computable in polynomial time using variants of Dijkstra’s shortest-path algorithm, pseudo-polynomial algorithms for computing cost-bounded reachability probabilities and related quantiles have been presented in [Ummels and Baier 2013; Baier et al. 2014]. Indeed, no algorithms with better complexity can be expected:

**Theorem 2.5** ([Haase and Kiefer 2015]). Deciding whether there is a strategy in an MDP $M$ such that the probability of cost-bounded properties exceeds a given probability threshold is PSPACE-hard, and PSPACE-complete for the case where $M$ is acyclic.

Conditional probabilities and expectations are an important concept, e.g., for a quantitative analysis of stochastic systems with rare event scenarios. Given an MDP $M = (S, Act, P)$, a state $s \in S$, and path properties $\varphi$ and $\psi$, we consider the problem of computing

$$
\Pr_{M,s}^{\max}(\varphi \mid \psi) \overset{\text{def}}{=} \max_{s' \in S} \Pr_{M,s}^s(\varphi \mid \psi) = \max_{s' \in S} \frac{\Pr_{M,s}^s(\varphi \land \psi)}{\Pr_{M,s}^s(\psi)}.
$$

Here, $\varphi$ is called the objective and $\psi$ the condition of the conditional probability, respectively. Minimal conditional probabilities are defined analogously, but one can safely restrict to the maximizing case as $\Pr_{M,s}^{\min}(\varphi \mid \psi) = 1 - \Pr_{M,s}^{\max}(\neg \varphi \mid \psi)$. In [Andrés 2011], a model-checking algorithm for MDPs and PCTL formulas extended by constraints for conditional probabilities has been presented, running in exponential time when both the objective and the condition are reachability properties. Using the reset method proposed in [Baier et al. 2014], the synthesis of maximizing strategies for conditional probabilities with reachability objective and condition can, however, be achieved in polynomial time. The latter approach has been generalized for $\omega$-regular properties, implemented in state-of-the art model checkers such as STORM [Dehnert et al. 2017] and PRISM [Märcker et al. 2017], and applied in several case studies. In [Baier et al. 2017a], an exponential-time algorithm for computing the maximal conditional expectation and a corresponding optimal strategies has been established. Recently, concepts for conditional probabilities and quantiles have been combined in computing conditional-value-of-risk for mean-payoff and reachability objectives in Markov chains and MDPs [Kretinsky and Meggendorfer 2018].

### 2.3.4. Robust systems: optimization under resilience constraints.

In the literature, there are many facets of describing the ability of some system to react on errors and environmental perturbations (see, e.g., surveys [Sterbenz et al. 2010; Attoh-Okine 2016]). We here concentrate on a few recently established aspects concerning synthesis problems to illustrate that this area is an active field of research, not limited to probabilistic systems but also important for non-probabilistic ones. The general idea behind most of the synthesis problems is to determine strategy decisions that increase resiliency of the system. Following [Sterbenz et al. 2010], resilience disciplines capture classical notions such as robustness, survivability, and fault tolerance. Robustness is a control-theoretic property relating the operation of a system to perturbations of its input. The task in robust reactive synthesis is to generate a controller guaranteeing an $\omega$-regular property under environment assumptions that might be violated temporarily (see, e.g., [Ehlers and Topcu 2014; Bloem et al. 2014]). In particular, [Bloem et al. 2014] considered quantitative notions of robustness formalized using mean payoff conditions. Robustness in (non-probabilistic) game structures has been considered, e.g., in [Huang et al. 2016], establishing a framework for synthesizing a strategy for one player in a safety game.
that maximizes the resilience under multiple (but within a fixed bound) disturbances. Similarly, resilient strategies that are optimal w.r.t. the number of disturbances covered were synthesized in [Neider et al. 2018], also using a game-theoretic approach that is not limited to safety games only. In [Majumdar et al. 2013] a formal definition of robustness w.r.t. \( \omega \)-regular properties using distance functions and an algorithm to synthesize robust strategies has been presented.

The synthesis of resilient strategies has not only been considered for non-probabilistic systems as in the aforementioned approaches, but also in the probabilistic setting, where probabilities occur, e.g., when modeling environmental disturbances or components such as sensors or within the synthesis of randomized resilient strategies. In [Dräger et al. 2015], a permissive controller synthesis framework has been introduced, generating multi-strategies to represent alternatives for controller decisions that minimize the penalties to be paid for “disallowed” actions. Their formalization relies on stochastic two-player games with probabilistic reachability constraints (via PCTL-like expressions) or constraints on the expected accumulated reward. The underlying decision problem is shown to be NP-hard and solvable using an incremental approach with mixed integer linear programming. Recently, resilient control strategies have been defined for MDP models [Baier et al. 2017b]. The authors call a strategy resilient when it ensures the recovery of the system under a given cost and probability constraint. They show that deciding whether there exists a resilient strategy is PSPACE-hard and can be done in pseudo-polynomial time. The synthesis of a resilient strategy that optimizes the long-run average reward under all resilient strategies is shown to be computable in pseudo-polynomial time using an LP-based approach.

3. FEATURE-ORIENTED SYNTHESIS

In feature-oriented system development (see, e.g., [Apel and Kästner 2009] for an overview), a feature describes a user-visible aspect or characteristics that may be present or absent in a system [Kang et al. 1990]. Features are most prominently used to model variability in (software) product lines [Clements and Northrop 2001], i.e., families of system variants that rely on a common code base and differ in the selection of features. Several extensions of the classical Boolean setting for features have been proposed in the literature. For instance, multi-features [Czarnecki et al. 2004] are features that may not only be absent or present in a variant but can appear multiple times, features with attributes [Cordy et al. 2013] are features that depend on (numerical) parameters, and dynamic features [Gomaa and Hussein 2003] are features that can be changed during run-time of the system. The classical setting for dynamic features amounts of simply activating or deactivating the feature, but also feature attributes or cardinalities might be subject of run-time changes for modeling adaptations within the system family. Beside others, these extensions enable feature-oriented concepts to be used also for a wide area of applications, e.g., to model and analyze adaptive and parametric systems.

The standard formalism for constructing variants in feature-oriented systems is provided by superimposition [Katz 1993], also apparent in delta-oriented programming [Schaefer et al. 2010]. For each feature it is specified which other features are required to be present and absent and how including the feature in a variant changes the behaviors of the system, i.e., what behaviors are removed, added, or modified. Opposed to the superimposition concept, featured transition systems provide a monolithic model to encode all behaviors of variants into a single model [Classen et al. 2013]. In [Dubsaff et al. 2015], a compositional formalism for dynamic feature-oriented systems has been introduced. The information about active features in a variant and their run-time switches is represented by an automata-based component called feature controller. Behaviors of features are encapsulated in separated components, composed with the fea-
ture controller using standard parallel-composition operations. These concepts naturally extend to probabilistic feature-oriented systems [Dubslaff et al. 2015] where feature components and controller are given by MDPs. In [Chrszon et al. 2018], the probabilistic framework of [Dubslaff et al. 2015] has been implemented in the tool PROFEAT, extending the input language of the prominent probabilistic model checker PRISM by feature-oriented concepts and supporting the family-based analysis of feature-oriented systems.

**Family-based analysis.** For the analysis of feature-oriented systems, one mainly distinguishes between two approaches: one-by-one and all-in-one analysis. Within a one-by-one analysis each variant is analyzed separately, while an all-in-one analysis uses a family model that encodes all variants in a single model and deduces the results for each variant from analyzing the family model in a single run. As the number of variants might be exponential in the number of features, a one-by-one analysis easily becomes infeasible for complex feature-oriented systems. Exploiting the fact that variants share behaviors of common features, symbolic representations of the family model might tackle the exponential blow-up apparent in one-by-one analysis approaches. As a consequence, all-in-one approaches are usually superior to one-by-one approaches, witnessed by several case studies in the literature (see, e.g., the survey [Thüm et al. 2014]).

### 3.1. Variant selection and featured strategy synthesis

The analysis of feature-oriented system usually asks for those variants that fulfill a given property, might it be functional (e.g., whether a safety property is fulfilled or not) or non-functional (e.g., whether a PCTL property is satisfied). In the context of probabilistic feature-oriented systems the task can be further extended to select optimal variants, e.g., asking for those variants maximizing the probability to reach a target or minimizing the expected energy consumption. Although not stated explicitly, the approach of [Ghezzi and Sharifloo 2013] can be used to select optimal variants following a given optimization criterion, using the parametric engine of PRISM to obtain rational functions for probabilities and expectations. This applies also to case studies following this approach, e.g., [Rodrigues et al. 2015; Lanna et al. 2018]. Using the approach by [Dubslaff et al. 2015], the selection of energy-optimal variants in heterogeneous tiled architectures were determined in [Baier et al. 2017b].

When dynamic features are involved in the system under consideration, another dimension regarding the selection of optimal variants arises. Then, also the synthesis of an optimal strategy how to activate and deactivate features during run-time is of interest. This synthesis problem has been considered in [Dubslaff et al. 2014] and [Dubslaff et al. 2015] for energy-aware server systems, extended towards synthesizing strategies that are optimal w.r.t. the energy-utility tradeoff expressed through energy-utility quantiles in [Klein et al. 2018].

### 3.2. Feature-oriented parameter synthesis

We now present another aspect in feature-oriented synthesis that has not yet been considered in the literature. Systems those behaviors depend on configurable parameters might be elegantly modeled through feature-oriented concepts including feature attributes for representing the system parameters. The arising system family then spans an optimization space over these parameters and one can exploit family-based analysis methods to synthesize optimal parameters. We exemplify the approach by
synthesizing energy-utility optimal parameters for the energy-aware network device EBOND [Hähnel et al. 2013].

EBOND. In the recent past, the focus of data-center design has seen a steady move from a purely performance-optimizing approach to an energy-aware approach due to cost, environmental, and infrastructural issues. For existing data-centers that do not yet include energy-aware hardware, [Hähnel et al. 2013] proposed a technology to adapt the usage of heterogeneous network interface cards (NICs). The test bed on which the authors evaluated their approach comprised a slow but low-energy 1 Gbit/s NIC (consuming between 1.41 W and 1.77 W) and a fast but high-energy 10 Gbit/s NIC (consuming between 7.86 W and 8.06 W). The EBOND network scheduling algorithm has three parameters: interval \( \Delta \) (in minutes), cooldown \( x \) (in minutes), and predictor \( y \in [0, 1] \cap \mathbb{Q} \). After each \( \Delta \) minutes, the requested average bandwidth \( b \) (in Gbit/s) within the last \( \Delta \) minutes and the current cooldown timer \( t \) is taken into account for deciding whether to switch the active NIC. In case \( b \cdot (1 + y) > 1 \text{ Gbit/s} \), the cooldown timer is reset to \( x \) and the 10 Gbit NIC is activated. Otherwise, the cooldown timer is decreased by \( \Delta \) and the 1 Gbit/s NIC is activated. In case \( x \leq 0 \). The authors considered three variants of the algorithm: balanced \( (x = 30 \text{ minutes and } y = 0.1) \), high savings \( (x = 0 \text{ minutes and } y = 0.1) \), and aggressive \( (x = 0 \text{ minutes and } y = 0) \). Using a simulation-based approach on real-word server data logs, they demonstrated that significant energy-savings can be achieved with these variants. The potential of saving energy mainly stems from the different bandwidth demands during day and night time. However, energy savings come at the cost of package delays, occurring when the bandwidth demands exceed 1 Gbit/s while on the slow NIC. Hence, there is a tradeoff between saving energy while guaranteeing utility in terms of service quality of the data center.

Feature-oriented model of EBOND. To model the EBOND protocol as a feature-oriented system, we rely on the framework presented in [Dubslaff et al. 2015; Chrszon et al. 2018]. The static parameter feature with feature attributes \( x \) (cooldown) and \( y \) (predictor) set the decision-making parameters. Two dynamic features 1Gbit and 10Gbit model the NICs, those switches are maintained by a deterministic feature controller component according to the EBOND protocol described above, making decisions depending on the feature attributes \( x \) and \( y \) of the parameter feature. The power measurements for both NICs from [Hähnel et al. 2013] are discretized with a granularity of 25 Gbit/s and included in the respective MDP-model of the features as non-negative weight function. A static environment feature with the feature attribute \( \Delta \) (interval) encapsulates the average bandwidth requirements in time steps of \( \Delta = 5 \) (minutes), obtained from statistical analysis of real-world server data logs: We exploit the known shift between day and night, convoluting the 42 days ubuntu-release server log from [Hähnel et al. 2013] into a probability distribution over one single day. As for the power measurements, a granularity of 25 Gbit/s is used. Figure 1 illustrates the bandwidth convolution, depicting the minimal (green), average (blue), and maximal (red) bandwidth demands occurring in the used server logs. We also highlighted the critical threshold at 1 Gbit/s for switching from the slow low-energy NIC to the fast high-energy NIC. Note that in the related case studies of [Dubslaff et al. 2014; Dubslaff et al. 2015; Klein et al. 2018] the bandwidth demands were modeled without using statistical data over real-word server logs.

Family-based synthesis of energy-utility optimal EBOND parameters. While the simulation-based approach of [Hähnel et al. 2013] allowed for the estimation of the energy consumption and the level of utility provided by the EBOND protocol for given parameters, it is not capable to reason about the energy-utility tradeoff and to synthesize energy-utility optimal parameters. We now show how this can be achieved us-
ing probabilistic model-checking techniques applied on the feature-oriented model of EBOND. To synthesize parameters $x$ and $y$ that provide optimal energy-utility tradeoff, we consider the attributes of the parameter feature, where the cooldown parameter $x$ ranges from 0 to 40 minutes in $\Delta$-steps and the predictor parameter $y$ ranges from 0 to 0.3 in $\frac{1}{30}$-steps. To this end, the EBOND-protocol family comprises 90 variants. We considered an energy-utility quantile (see Section 2.3.3) over one day, i.e., the set of target states end_of_day is determined by those states where the time bound of 288 $\Delta$-steps is reached. As proposed in [Baier et al. 2014], the utility measure $u$ is encoded into the state space and corresponds to the number of $\Delta$-steps where no package delay occurred. The utility constraint utility_goal is given by the set of states where $u \geq 276$, i.e., in at most 12 time slots of $\Delta = 5$ minutes, a package delay occurred. Let us define a goal function $q: \{0, 5, \ldots, 40\} \times \{0, 0.03, 0.06, \ldots, 0.3\} \times \{0, 1\} \cap \mathbb{Q} \rightarrow \mathbb{Q}$ by the following energy-utility quantile values w.r.t. a probability bound $p$:

$$q(x, y, p) \leftrightarrow \min \{ e \in \mathbb{Q} : \Pr_{x,y}(\Diamond \leq e (\text{end_of_day} \land \text{utility_goal})) > p \} .$$

Here, $\Pr_{x,y}(\cdot)$ denotes the probability measure of the variant with parameters $x$ and $y$. Given $p \in [0, 1] \cap \mathbb{Q}$, the EBOND parameter-synthesis problem now amounts to find a parameter combination that minimizes $q(\cdot, \cdot, p)$, i.e., when the probability threshold is provided through the service contract on the data center the EBOND technology is applied on, we ask for the parameter combination that guarantees the energy-utility tradeoff with a minimal amount of energy. We solved the EBOND parameter-synthesis problem for the probability threshold $p=1$, as the algorithm behind quantitative quantile computations [Baier et al. 2014] is conceptional able to report on intermediate probability bounds and their quantile values. On the right of Figure 1, the results of the family-based analysis are shown for $p = 0.95$, where each plot corresponds to a fixed value of the predictor parameter $y$. Following our optimization criterion, one can easily determine that the parameter combination $x = 20$ minutes and $y = 0$ is optimal, guaranteeing the required level of service quality with a daily energy investment of 107.43 Wh. This illustrates that the predictor parameter $y$ does not have a big impact on the utility-level guarantees and increasing the cooldown time $x$ is more effective.
We carried out\textsuperscript{3} both, a one-by-one and an all-in-one analysis of the EBOND feature-oriented system to solve the EBOND parameter-synthesis problem using the semi-symbolic SPARSE engine of the model checker PRISM [Kwiatkowska et al. 2011] with variable reordering enabled [Klein et al. 2018]. Table I reports on the statistics of the experiments, indicating the number of states of the resulting model, the number of MTBDD nodes\textsuperscript{4} used for the symbolic representation of the model, and the timings of the analysis. In case of the one-by-one approach, the values correspond to the sum of the characteristics of all analysis runs on single variants. In the number of nodes one can observe that the family model indeed exploits the shared behaviors of common features, e.g., the comparably sophisticated environment feature. The analysis time is more than four times shorter in case of the all-in-one analysis.

4. SYNTHESIS OF PROBABILITY PARAMETERS

Results of the formal analysis and synthesis with Markovian models and quantitative specifications crucially depend on the concrete transition probabilities. Even small perturbations of the probability values can affect the analysis or synthesis results. This can be problematic in cases where only estimates of the transition probabilities are available. This, for instances, applies to cases where probability values in the models are derived using statistical or learning methods.

This motivated the introduction of Markov chains where intervals are attached to the transitions rather than concrete transition probabilities [Jonsson and Larsen 1991; Sen et al. 2006; Chatterjee et al. 2008; Delahaye et al. 2011; Caillaud et al. 2013]. Two semantics have been introduced for interval-valued Markov chains. The uncertain semantics treats interval-valued Markov chains as families of Markov chains with the same state space and transition probabilities in the intervals. The nondeterministic semantics considers interval-valued Markov chains as MDPs where the concrete probability values are chosen nondeterministically.

Parametric Markov chains can be seen as a generalization of interval-valued models with the uncertain semantics. Instead of intervals of potential probability values, they use polynomial functions over a fixed set of parameters to specify the transition probabilities [Daws 2005; Lanotte et al. 2007; Hahn et al. 2011b]. Such parametric models can be seen to define a family of concrete probabilistic models that arise by plugging in concrete values for the parameters.

Formally, a parametric Markov chain with parameters $x_1,\ldots,x_k$ is a tuple $\mathcal{M}[x_1,\ldots,x_k] = (S,E,Y)$ where $(S,E)$ is a finite directed graph, i.e., $S$ is a finite state space and $E \subseteq S \times S$ specifies the transitions, $Y: E \to \mathbb{Q}[x_1,\ldots,x_k]$ a function that assigns to each transition $s \to s'$ a polynomial over $x_1,\ldots,x_k$ with rational coefficients. A parameter valuation $\xi: \{x_1,\ldots,x_k\} \to \mathbb{R}$ assigning real numbers to the parameters is said to be admissible if $\sum_{s' \in E(s)} Y(s,s')(\xi_1,\ldots,\xi_k) = 1$ for each non-trap state $s \in S$ where $E(s) = \{s' \in S : (s,s') \in E\}$. The (concrete) Markov chain given by an admissible parameter valuation $\xi$ is $\mathcal{M}_\xi = (S,P_\xi)$ where $P_\xi(s,s') = Y(s,s')(\xi_1,\ldots,\xi_k)$ if $(s,s') \in E$ and $P_\xi(s,s') = 0$ if $(s,s') \notin E$. Then, the semantics of $\mathcal{M}[x_1,\ldots,x_k]$ is the family of (concrete) Markov chains induced by the admissible parameter valuations.

On the left of Figure 2 a parametric Markov chain is depicted, describing a parametric variant of Knuth and Yao’s protocol for simulating a six-sided dice by coin-flipping.

3\textsuperscript{Hardware setup: Intel Xeon E5-2680@2.70GHz, 128 GB RAM; Turbo Boost and HT enabled; Debian GNU/Linux 9.1}

4\textsuperscript{PRISM uses multi-terminal binary decision diagrams (MTBDDs) [Hermanns et al. 2003] for its symbolic engines.}
with coefficients represented as polynomials derived from $\gamma$, $p$ is the probability vector, and $b$ a column vector. Thus, we can view $A$ as a matrix over the field $\mathbb{Q}(x_1, \ldots, x_k)$ of rational functions with parameters $x_1, \ldots, x_k$ and then apply Gaussian elimination (or the related state-elimination approach known for generating regular expressions for finite automata) to compute the solution vector $p$. This approach has been first proposed in [Daws 2005; Lanotte et al. 2007] and later refined using computer-algebra tools to simplify the rational functions obtained in intermediate steps [Hahn et al. 2011b] or using decompositions into strongly connected components and factorization techniques for polynomials [Jansen et al. 2014]. These techniques have been implemented in the tools PARAM [Hahn et al. 2010], its reimplemention in PRISM [Kwiatkowska et al. 2011], and in STORM [Dehnert et al. 2017]. More recently, it has been observed that the expensive computations of greatest common devisors of polynomials can be avoided using one-step fraction-free Gaussian elimination [Hutschenreiter et al. 2017]. The latter yields:

**Theorem 4.1** ([Hutschenreiter et al. 2017]). The rational functions for reachability probabilities in parametric Markov chains are computable in time $\mathcal{O}(\text{poly}(n, d)^k)$ where $n$ is the number of states, $d$ the maximal degree of the polynomials in $\gamma$ and $k$ the number of parameters.

The same holds for expected accumulated weights or the expected mean payoff. Thus, in the univariate case $k=1$ (or for any fixed number $k$ of parameters), the rational functions are computable in polynomial time.

The applications of parametric Markov chains and the algorithms to compute rational functions for reachability probabilities or expected values are manyfold. These include the analysis of systems with unknown transition probabilities where the task is a family-based analysis to provide guarantees for all admissible parameter valuations, but also the treatment of systems with tiny transition probabilities, which, e.g., applies often to models that capture the effect of very exceptional errors and where standard analysis techniques for the analysis fail due to numerical problems.

The parameter-synthesis problem goes one step further and asks to find parameter valuations where a set probability or expectation constraints holds, possibly in combination with other side constraints. For a simple example, we regard the Markov chain shown in Figure 2. While for a fair coin, $x_1 = x_2 = x_3$, the result of the outcome “six” is 1/6, one might ask how to manipulate the coins such that the probability for “six” is 1/2. Assuming the same coin is used in all rounds (i.e., $x_1 = x_2 = x_3$), the task is to find a value $p$ such that $(-p^3 + 3p^2 - 3p + 1)/(p^3 - p + 1) = 1/2$. With Newton’s method we obtain $p \approx 0.26102$. While this is a toy example, several important applications of the parameter-synthesis problem have been studied in the literature. Let us mention a few of them.
The use of parametric model checking for supporting decision making at run-time in adaptive software has been investigated by several authors, see, e.g., [Calinescu et al. 2012; Filieri et al. 2016; Su et al. 2016b]. The idea is to precompute the rational functions for a set of quantitative properties, which can be efficiently evaluated at run-time for guiding software adaptions when the environment changes. An iterative decision-making scheme for MDPs with interval estimates for the transition probabilities has been proposed in [Su et al. 2016a]. This approach serves to tackle the trade-off between accuracy, data usage and computational overhead. It successively computes strategies that are optimal for a cost-bounded reachability condition and checks their confidence optimality. If not, the iteration returns to data sampling.

Besides supporting decisions at run-time, the parametric approach can also be very useful to find (nearly) optimal system configurations at design-time. For example, [Aflaki et al. 2017] employs parametric model-checking techniques to find probability distributions for a probabilistic self-stabilizing protocol that achieves minimum average recovery time, while [Leuschner et al. 2017] uses the parametric approach to determine (an approximation of) the optimal frequency of periodic reboots for an inter-process communication protocol of a space probe where optimality is understood with respect to the long-run availability. In [Bartocci et al. 2011] the model-repair problem is considered where some transition probabilities of a Markov chain are declared to be controllable and the task is to modify the controllable probabilities such that a PCTL property holds for the modified Markov chain and the costs for deriving the new Markov chain are minimal. The latter is formalized using a nonlinear cost function, which leads to a nonlinear programming problem and is shown to be related to the optimal-controller synthesis problem for discrete linear dynamical systems. An alternative model-repair approach using parametric Markov chains and greedy methods has been proposed in [Pathak et al. 2015].

Beyond parametric Markov chains, several authors have addressed the parametric setting to different and more expressive stochastic models. This includes methods to synthesize parametric rate values in continuous-time Markov chains that ensure the validity of bounded reachability properties [Han et al. 2008]. Timeout-synthesis problems for continuous-time stochastic models have been investigated, e.g., in [Brázdil et al. 2015; Baier et al. 2017a]. The parameter-synthesis problem for interval Markov chains with parametric lower and upper endpoints of intervals has been addressed in [Bart et al. 2018] where methods to find concrete values for the parameters such that the resulting interval-valued Markov chain maximizes the probability for a reachability condition have been proposed. Parametric approaches for MDPs have been studied in [Hahn et al. 2011a; Cubuktepe et al. 2017; Cubuktepe et al. 2018]. Another recent approach to deal with unknown transition probabilities uses online learning techniques, e.g., applied on MDP models in [Kreținăský et al. 2018], where only the support of distributions is known in advance and where the task is to synthesize a strategy that almost surely satisfies a parity condition and is nearly optimal w.r.t. a mean-payoff objective.

5. CONCLUSION

In this article we gave a brief overview on traditional and recent directions for synthesis problems based on MDPs with manyfold application areas. The first part reports on synthesis problems of the classical type, i.e., synthesizing strategies, but with non-standard objectives. A common concept of the last two sections focused on family-based analysis that supports to determine system instances (family members) satisfying certain constraints. This article is far from being complete and there are several other directions not mentioned so far. One direction are strategy-synthesis problems for partially observable MDPs, POMDPs for short, [Papadimitriou and Tsitsiklis...
The challenge here is that strategies do not have access to the full history. Many verification problems are conceptually close to decision problems for probabilistic finite automata and therefore undecidable. This, for instance, applies to algorithmic questions for expected costs [Madani et al. 2003], but also to verification problems for infinite-horizon properties under qualitative probability thresholds [Baier et al. 2012] or for distributed strategies [Giro and D’Argenio 2007]. To escape from undecidability, one can restrict to finite-horizon properties (see, e.g., [Mundhenk et al. 2000]) or can impose syntactic restrictions on the type of strategies. For instance, EXPTIME-completeness has been established for finite-memory strategies and parity objectives and almost-sure satisfaction in POMDPs [Chatterjee et al. 2016a]. Further, the stochastic shortest-path problem for POMDPs with positive cost functions was shown to be solvable in double exponential time [Chatterjee et al. 2016b]. Decidability has also been established for restricted classes of strategies in probabilistic distributed systems [Giro et al. 2009]. Another very active research area is the strategy synthesis in stochastic two- or multi-player games (see, e.g., [Condon 1992; de Alfaro and Majumdar 2004; Chen et al. 2013; Chatterjee et al. 2013a; Chatterjee and Doyen 2016; Bertrand et al. 2017]).

REFERENCES


This installment of the conference report column includes reports on two recent events related to SIGLOG:

— Agata Ciabattoni, Björn Lellmann, and Kees van Berkel report on the workshop “Deontic Reasoning: from Ancient Texts to Artificial Intelligence”, which took place at the Vienna University of Technology (TU Wien) on June 11-13, 2018 (see https://mimamsa.logic.at/atai/).

— Daniele Ahmed, Katherine Fletcher, and Julian Gutierrez report on the 2018 Federated Logic Conference (FLoC 2018), which was hosted by Oxford University on July 6-19, 2018 (see http://www.floc2018.org).

I am most grateful to all the authors for their detailed reports!

As usual, I look forward to receiving your reports and/or personal impressions on workshops, conferences, summer schools and scientific meetings broadly related to SIGLOG. I will also be pleased to hear all your ideas and suggestions for future installments of the column.
The interdisciplinary workshop 'Deontic Reasoning: from Ancient Texts to Artificial Intelligence' (ATAI), bringing together experts from the fields of Logic, Sanskrit, Philosophy, Artificial Intelligence and Law, was held at the Vienna University of Technology (TU Wien) on June 11-13, 2018. In nuce, the aim of the workshop was to foster new connections between the aforementioned research areas and facilitate the interchange of ideas with respect to shared grounds of interest, in particular normative reasoning.

The workshop was part of the research project 'Reasoning Tools for Deontic Logic and Applications to Indian Sacred Texts' (2017-2021) funded by the Wiener Wissenschafts-, Forschungs- & Technologiefonds Vienna (WWTF). The project aims to use formal logic to analyze and provide a better understanding of the deontic reasoning of the Māṇḍūkya school of Indian philosophy. Flourishing for more than two millennia, Māṇḍūkya provided a systematic and formal interpretation of the normative part of the Indian Sacred Texts, the Vedas. To this aim Māṇḍūkya established interpretative principles which are so rational and systematic that some of them are still applied in Indian Jurisprudence to decide court cases. The overarching aim of the workshop was to brainstorm and explore potential mutual benefits of the application of formal tools to the analysis of ancient texts. Throughout the workshop the following questions functioned as guiding threads:

i) How can formal tools, such as mathematical logic or argumentation frameworks, enhance the understanding of ancient texts?

ii) What can be learned from ancient analyses with respect to formal reasoning about normative statements (e.g. for ethical machines such as self-driving cars)?

The ATAI workshop served to map out these benefits in a lively interdisciplinary setting.

The three-day event was attended by 32 participants from 14 different universities. The scientific part of the workshop consisted of 12 one-hour talks, given by invited speakers, together with 5 panel discussions. The presentations were grouped by five central themes generally addressed through the topics of the talks. Each of these blocks closed with a corresponding discussion, paneled by the respective speakers and moderated by members of the organization committee.

**Featured talks**

- Karin Preisendanz (University of Vienna)
Opening Talk

» Agata Ciabattoni (TU Wien), Elisa Freschi (ÖAW, University of Vienna)
  Deontic Reasoning: From Mīmāṃsā to AI

» Matthias Baaz (TU Wien)
  Logical Aspects of Legal Reasoning

» Giovanni Sartor (University of Bologna)
  Defeasible Legal Argumentation

» Dov Gabbay (King’s College London)
  Principles of Talmudic Logic - Sample Export to Modern AI

» Lawrence McCrea (Cornell University)
  Contextual Factors in the Interpretation of Prohibitions

» David Brick (Yale University)
  Arguments Regarding Sati from Classical Hindu Law

» Xavier Parent (University of Luxembourg)
  A Rule-Based Deontic Reasoner

» Parimal Patil (Harvard University)
  The Cognition of Commands in Navya-Nyāya

» Eberhard Guhe (Fudan University)
  Ross’ Paradox and the Navya-Nyāya Interpretation of Injunctions

» Patrick Cummins (Cornell University)
  Appointment as Linguistic Category in Prabhākara’s Hermeneutics of Deontology

» Andrew Ollett (University of Chicago)
  Different Deontic Concepts in Mīmāṃsā

In the remainder of this report we briefly discuss a selection of talks and corresponding panel discussions, that explicitly address the interdisciplinary potential of combining logic and AI with law and with ancient texts.

With respect to normative reasoning in legal systems and its formal counterpart, Matthias Baaz spoke about how different contemporary legal systems deal with logical contradictions and presented several commonly applied schemata of legal reasoning that are particularly challenging for formal logic. Giovanni Sartor gave a tutorial on reasoning in legal argumentation, including several attacking strategies for rebutting
and undercutting juridical arguments. From the Indological side, David Bricks dis-
cussed various historical solutions and arguments concerning the problem of Sati, i.e.,
the immolation of a widow on her deceased husband's pyre, in Hindu law.

During the corresponding panel discussion, there was a vivid debate concerning the
rôle of contradictions in normative systems: are they undesirable and should they be
resolved? The discussion revealed that several scientific fields employ similar (formal)
answers: From a legal perspective it was emphasized that, although in some juridical
systems contradictions are resolved by adjusting the system, others leave contradic-
tions untouched and, instead, resolve conflicting cases through principles such as lex posterior
derogat legi priori (cf. ‘specificity principle’ in AI). In accordance with this, from the side
of Indian Philosophy, it was pointed out that Mimânsâ authors aimed at resolving any
apparent normative conflict through very similar interpretative principles, since the
Vedas, as the source of normative statements, is always consistent. In particular, the
Mimânsâ authors also reasoned by specificity, and, as a very last resort, would apply
the principle of Vikalpa: ‘in case of two conflicting obligations, you ought to perform at
least one obligation’ (cf. ‘disjunctive response’ in deontic logic).

Concerning the connection between logic, AI and the study of ancient philosophical
texts, Dov Gabbay gave an introduction to the project of extracting different systems
of Talmudic Logic. This included discussions of the Talmudic interpretation of future
conditionals, changes of identity over time, and how Talmudic reasoning suggests a new
approach to the paradox of the heap by viewing properties of an object as dependent on
how it is constructed. In a similar spirit, Eberhard Guhe presented his formalisation
of obligations and permissions in the interpretation of another fundamental school of
Indian Philosophy called Navya Nyâya. Witnessing the potential of cross-fertilisation of
the different disciplines, his formalisation provides a novel method for solving several
well-known paradoxes of deontic logic, such as Ross’ Paradox or Veltman’s Puzzle.
Conversely, the benefits of applying formal methods to the study of ancient texts were
demonstrated by Andrew Ollett in his presentation of a formalisation of normative
ccepts in the Mimânsâ interpretation.

Questions concerning benefits of the formalisation of philosophical notions were also
considered in detail in the panel discussions. Among the many points discussed, it
was stressed by Gabbay that the study of decision making in AI must be able to deal
with the reasoning employed by real people and not just with idealised mathematical
reasoning. Yet, employing human reasoning is exactly what is essential to systems like
Talmudic or Mimânsâ reasoning, which makes their study very relevant for designing
good AI systems. Furthermore, it was generally agreed upon that hermeneutics and
philosophical reasoning have more aspects than just logic, implying that logic is not
suitable for all philosophical problems. However, as a benefit for philosophy, it was
also pointed out that formal logic can be used as a tool for deriving new (philosophical)
hypotheses and questions, even explaining or solving certain conflicts.

The workshop took place in the Zemanek room at the TU Wien. The venue was
in close proximity to the historical city centre of Vienna with its many sights. The
attendants enjoyed ample time for discussion over original Sicilian coffee and sweets
during the extended coffee breaks. The event was funded by the WWTF project, Uni-
versity of Vienna, Vienna Center for Logic and Algorithms, Institut für Kultur- und
Geistesgeschichte Asiens, TU Wien and Wolfgang Pauli Institute.

Further information about the workshop can be found on the official website under
[https://mimamsa.logic.at/atai/](https://mimamsa.logic.at/atai/)
Federated Logic Conference (FLoC) at Oxford

2018 FLoC Review and Impressions.

Oxford University was proud to host the Federated Logic Conference (FLoC), in July of 2018. FLoC is a meta-conference, which occurs every 4 years, with the 2018 edition bringing together 9 major conferences in the fields of logic and foundations of computer science. With 9 headline conferences, nearly 80 workshops, a summer school, a public lecture and a public debate in the Oxford Union (and a variety of social events), FLoC 2018 hosted over 2000 individuals between 30 June and 19 July. The stars of the show were the 9 lead conferences: CAV, CSF, FM, FSCD, ICLP, IJCAR, ITP, LICS and SAT.

FLoC provides a unique environment for research and networking to flourish: attendees are welcome to hop between parallel conferences and workshops. An army of 160 volunteers made it all possible and we are deeply grateful to them all (many of our volunteers were students, but even the Heads of HR and Administration for the Computer Science Department contributed their time!). We believe all of these efforts really paid off, as every attendee seemed to have enjoyed the event, which was planned in every detail. As part of the volunteering team and scheme, we could see first-hand how the overall organization was very well handled by the main organisers: Daniel Kroening, Marta Kwiatkowska, and Moshe Vardi. They were greatly supported by a huge number of local and conference organizers, which made it possible that every conference day was run smoothly. In particular, as part of the Volunteering team, we had various roles to play and events to oversee: registering and welcoming participants, guiding them to other buildings, directing them to talks and other duties to maintain maximum safety throughout. With nearly 90 events happening in such a short period of time, FLoC brought a huge number of great results in many areas of logic and computer science. FLoC, as expected, was an incredibly challenging endeavour. The whole event took place in two main sites, the Blavatnik School of Government and the Mathematical Institute, where all conferences and most workshops were held. Some workshops also took place in the Department of Computer Science and in two beautiful Oxford colleges, St Anne’s College and Green Templeton College. Lunch, tea breaks, and refreshments were served in all venues as well as in the nearby Freud cafe and bar. In the end, more than 20 rooms and lecture theatres were used to accommodate about 2000 participants who visited the University of Oxford during FLoC.
One of the best parts of being at FLoC was to find exciting areas of research, different from those people were interested in or focusing on: FLoC participants could attend any talk and the speakers made their work really interesting and involving. Every day there were a vast number of talks, which made it hard to choose where to go! There were great talks from both academia and industry so that everyone had the opportunity to see how, why and who worked on the latest technologies and challenges in computing. There were plenaries and keynotes by amazing speakers, Turing award winners, professors, researchers and students. FLoC now fosters a mature and healthy community of academics and industrialists. Even though FLoC brings together many academic events sharing some research interests, the variety of topics that are presented at FLoC can please most researchers in computer science. The main topics of research included work in verification, computer security, models of computation, formal methods, semantics and programming languages, automated reasoning, theorem proving, artificial intelligence and machine learning, and, of course, the logical foundations of computing. With such a wonderfully wide range of research developments to be presented, you will definitely want to save your days for the next FLoC!

For more information about FLoC 2018 at Oxford, see www.floc2018.org. Recordings of the Summer School, keynote and plenary lectures, public lecture and debate are all available online on the FLoC 2018 YouTube channel: http://goo.gl/GkkZjM
SIGLOG MONTHLY 201

DANIELA PETRİŞAN, Université Paris Diderot

SIGLOG Monthly 201
October 19, 2018

******************************************************************************
* 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

* NEWS
  LICS 2019 - Call for Papers
  LICS 2019 Call for Workshop Proposals
  WINNERS OF THE 2018 ALONZO CHURCH AWARD
  ACM SIGLOG Announcement
  EATCS Bulletin - Call for abstracts

* DEADLINES
  Forthcoming Deadlines

* CALLS
  FSEN 2019 - Third Call for Papers
  ETAPS 2019 - Joint Call for Papers
  HELMUT VEITH STIPEND - Call for Applications
  FM'19 - Call for Workshop and Tutorial Proposals
  SPECIAL ISSUE OF AIJ ON EPISTEMIC PLANNING - Call for Papers
  JELIA 2019 - First Call for Papers
  NLPinAI 2019 - Call for papers
  CiE 2019 - First Call for Papers
  FSCD 2019 - First Call for Papers
  EPIT 2019 - Announcement
  TACL 2019 - Early announcement
  CATEGORY THEORY 2019 - Early announcement
  RSSRail 2019 - Call for Papers
  FM 2019 - First Call for Papers

* JOB ANNOUNCEMENTS
  PHD STUDENT POSITION IN DATABASE THEORY AND LOGIC
  AI*IA INCOMING AND OUTGOING MOBILITY GRANTS 2018 - CALL FOR RESEARCH VISITS
  POSTDOCTORAL RESEARCHER IN LOGIC, MILAN
34TH ANNUAL ACM/IEEE SYMPOSIUM ON LOGIC IN COMPUTER SCIENCE (LICS 2019)

Call for Papers
24-27 June 2019, Vancouver
https://lics.siglog.org/lics19/
Full Papers Due: 11 January 2019

* SCOPE
The LICS Symposium is an annual international forum on theoretical and practical topics in computer science that relate to logic, broadly construed. We invite submissions on topics that fit under that rubric.

* Suggested, but not exclusive, topics of interest include: automata theory, automated deduction, categorical models and logics, concurrency and distributed computation, constraint programming, constructive mathematics, database theory, decision procedures, description logics, domain theory, finite model theory, formal aspects of program analysis, formal methods, foundations of computability, games and logic, higher-order logic, lambda and combinatory calculi, linear logic, logic in artificial intelligence, logic programming, logical aspects of bioinformatics, logical aspects of computational complexity, logical aspects of quantum computation, logical frameworks, logics of programs, modal and temporal logics, model checking, probabilistic systems, process calculi, programming language semantics, proof theory, real-time systems, reasoning about security and privacy, rewriting, type systems and type theory, and verification.

* IMPORTANT DATES
Authors are required to submit a paper title and a short abstract of about 100 words in advance of submitting the extended abstract of the paper. The exact deadline time on these dates is given by anywhere on earth (AoE).

Titles and Short Abstracts Due: 4 January 2019
Full Papers Due: 11 January 2019
Author Feedback/Rebuttal Period: 4-8 March 2019
Author Notification: 29 March 2019

* KLEENE AWARD FOR BEST STUDENT PAPER: An award in honour of the late Stephen C. Kleene will be given for the best student paper(s), as judged by the program committee.

* SPECIAL ISSUES: Full versions of up to three accepted papers, to be selected by the program committee, will be invited for submission to the Journal of the ACM. Additional selected papers will be invited to a special issue of Logical Methods in Computer Science.

34TH ANNUAL ACM/IEEE SYMPOSIUM ON LOGIC IN COMPUTER SCIENCE (LICS 2019)

Call for Workshop Proposals
https://lics.siglog.org/lics19/

* The Thirty-Fourth Annual ACM/IEEE Symposium on Logic In Computer Science (LICS’19) will be held in Vancouver, Canada on June 24-27,
2019. The workshops will take place on June 22-23, 2019.

* Researchers and practitioners are invited to submit proposals for workshops on topics relating logic - broadly construed - to computer science or related fields. Typically, LICS workshops feature a number of invited speakers and a number of contributed presentations. LICS workshops do not usually produce formal proceedings. However, in the past there have been special issues of journals based in part on certain LICS workshops.

* Proposals should include:
  - A short scientific summary and justification of the proposed topic. This should include a discussion of the particular benefits of the topic to the LICS community.
  - A discussion of the proposed format and agenda.
  - The proposed duration, which is typically one day (two-day workshops can be accommodated too).
  - Procedures for selecting participants and papers.
  - Expected number of participants. This is important for the room!
  - Potential invited speakers.
  - Plans for dissemination (for example, special issues of journals).

Proposals should be sent to Frederic Blanqui: frederic.blanqui@inria.fr

* IMPORTANT DATES
  - Workshop Proposal Submission deadline: November 15, 2018
  - Notification: December 1, 2018
  - Program of the workshops ready: May 19, 2019
  - Workshops: June 22-23, 2019
  - LICS conference: June 24-27, 2019

* The workshops selection committee consists of the LICS General Chair, LICS Workshops Chair, LICS 2019 PC Chair and LICS 2019 Conference Chairs.

WINNERS OF THE 2018 ALONZO CHURCH AWARD

* The 2018 Alonzo Church Award for Outstanding Contributions to Logic and Computation is given jointly to Tomas Feder and Moshe Y. Vardi for fundamental contributions to the computational complexity of constraint-satisfaction problems. Their contributions appeared in two papers:

* CONTRIBUTION SUMMARY: The Feder-Vardi project aimed at finding a large subclass of NP that exhibits a dichotomy (all problems are either in PTIME or NP-complete). The approach is to find this subclass via syntactic prescriptions. The paper identified a class of problems specified by "monotone monadic SNP without inequality", which may exhibit this dichotomy. Feder and Vardi justified placing all three restrictions by showing, using Ladner's theorem, that classes obtained by using only two of the above three restrictions do not show this dichotomy. They then explored the structure of this class. They show that all problems in this class reduce to the
seemingly simpler class CSP -- Constraint Satisfaction Problems. They divided CSP into subclasses and tried to unify the collection of all known polytime algorithms for CSP problems and extract properties that make CSP problems NP-hard. They conjectured that the class CSP (and therefore, also MMSNP) also satisfy the dichotomy property. This became known as the Feder-Vardi Dichotomy Conjecture. The Dichotomy Conjecture stimulated an extensive research program, which culminated in 2017 in two independent proofs, by A. Bulatov and by D. Zhuk, of its correctness.

ACM SIGLOG ANNOUNCEMENT
http://siglog.acm.org
* The ACM has recently chartered a Special Interest Group on Logic and Computation (ACM SIGLOG).
* We are pleased to announce the 2016 ACM SIGLOG election results for the term of 1 July 2016 - 30 June 2019. The SIGLOG Chair is Prakash Panangaden and the other officers are Luke Ong (vice-Chair), Amy Felty (Treasurer) and Alexandra Silva (Secretary).
* The ACM-IEEE Symposium on Logic in Computer Science is the flagship conference of SIGLOG. SIGLOG will also actively seek association agreements with other conferences in the field. A SIGLOG newsletter (SIGLOG News) is also 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).

EATCS Bulletin - Call for abstracts
* EATCS Bulletin, http://eatcs.org/index.php/eatcs-bulletin, has a section for "Technical contributions." To stimulate this section further, we have recently started publication abstracts of works that have been accepted by journals and/or conferences or have appeared in major archives. The topics of interest include all areas of theoretical computer science (for instance, see topics of the three Tracks of ICALP, http://www.easyconferences.eu/icalp2016/cfp.html).
* Abstracts should be rather detailed, 2-3 pages long in the format given at http://eatcs.org/index.php/eatcs-bulletin. Submissions should include the information on the full paper (the name of conferences, archives, etc) and sufficiently detailed explanation of its merits, e.g., importance, motivations, clear comparison with existing results and novelty and/or new ideas of proof techniques.
* The Bulletin is published in Feb, Jun and Oct. The deadline for the abstract submission is 15th of the previous month, for instance, Jan 15, 2018 for the Feb issue of 2018. All materials including tex and pdf files should be sent electrically to bulletin@eatcs.org and iwama@kuis.kyoto-u.ac.jp. Acceptance/rejection, decided based on
its merit mentioned above, will be notified as soon as possible. The Bulletin will not require copy-right transfer for accepted abstracts.

DATES
* FSEN 2019
  Third Call for Papers
  http://fisen.ir/2019
  Tehran, Iran, May 1-3, 2019
  IFIP Supported Event (IFIP WG 2.2 and IFIP TC2)
  http://www.ifip.org/
  Paper Submission: October 28, 2018 (AoE)
* ETAPS 2019
  Joint Call for Papers
  http://etaps.org/2019/call-for-papers
  Prague, Czech Republic, 6-11 April 2019
  Abstracts due: 9 November 2018 23:59 AoE
  Papers due: 16 November 2018 23:59 AoE
* HELMUT VEITH STIPEND
  Call for Applications
  For female MSc students in CS
  TU Wien
  Deadline: November 30, 2018.
* FM'19
  Call for Workshop and Tutorial Proposals
  Porto, Portugal, October 7-11, 2019
  formalmethods2019.inesctec.pt
  Deadline for proposals: November 16, 2018
  Notification of decision on workshops and tutorials: November 23, 2018
* SPECIAL ISSUE OF AIJ ON EPISTEMIC PLANNING
  Call for Papers
  Submission deadline: December 1, 2018
* JELIA 2019
  First Call for Papers
  Rende, Italy, May 8-10, 2019
  https://jelia2019.mat.unical.it/
  Paper submission deadline: 03 December 2018 (23:59 UTC-12)
* NLPinAI 2019
  Call for papers
  19 - 21 February, 2019, Prague, Czech Republic
  http://www.icaart.org/NLPinAI.aspx
  Paper Submission: December 20, 2018
* CiE 2019
  First Call for Papers
  Durham, United Kingdom
  July 15 - July 19, 2019
  https://community.dur.ac.uk/cie.2019/
  http://www.computability.org.uk
  Deadline for article submission: 14 January 2019 AOE
* FSCD 2019
  First Call for Papers
  24 - 30 June 2019, Dortmund, Germany
http://fscd-conference.org/
Full Papers Deadline: 11 February 2019

* EPIT 2019
  Announcement
  April 8-12, 2019, CIRM, Marseille Luminy, France
  Webpage of the event: https://conferences.cirm-math.fr/1934.html

* TACL 2019
  Early announcement
  Conference: June 17 - 21, 2019 in Nice
  School: June 10 - 15, 2019 in Île de Porquerolles
  https://math.unice.fr/tacl/2019/

* CATEGORY THEORY 2019 (CT 2019)
  Early announcement
  University Of Edinburgh, 7-13 JULY 2019
  http://conferences.inf.ed.ac.uk/ct2019/

* RSSRail 2019
  Call for Papers
  June 4-6, 2019, Lille, France
  https://conferences.ncl.ac.uk/rssrail2019/
  abstract submission deadline: January 5, 2019

* FM 2019
  First Call for Papers
  Porto, Portugal, October 7-11, 2019
  http://formalmethods2019.inesctec.pt/
  Full paper submission: 11 April, 2019, 23:59 AoE

EIGHTH INTERNATIONAL CONFERENCE ON FUNDAMENTALS OF SOFTWARE
ENGINEERING 2019 - THEORY AND PRACTICE (FSEN ’19)
Third Call for Papers
http://fsen.ir/2019
Tehran, Iran, May 1-3, 2019
IFIP Supported Event (IFIP WG 2.2 and IFIP TC2)
http://www.ifip.org/
* The topics of interest cover all aspects of formal methods,
especially those related to advancing the application of formal
methods in the software industry and promoting their integration
with practical engineering techniques.
* IMPORTANT DATES
  Abstract Submission (optional): October 19, 2018 (AoE)
  Paper Submission: October 28, 2018 (AoE)
  Notification: December 18, 2018
  Final pre-Conference Version: January 20, 2019 (AoE)
  Conference: May 1-3, 2019

* KEYNOTE SPEAKERS
  Rocco De Nicola, IMT School for Advanced Studies Lucca, Italy
  Giovanna Di Marzo Serugendo, University of Geneva, Switzerland
  Martin Wirsing, LMU Munich, Germany

* PROGRAM CHAIRS
  Hossein Hojjat - Rochester Institute of Technology, USA
  Mieke Massink - CNR-ISTI Pisa, Italy
22ND EUROPEAN JOINT CONFERENCES ON THEORY AND PRACTICE OF SOFTWARE (ETAPS 2019)
Joint Call for Papers
http://www.etaps.org/2019
Prague, Czech Republic, 6-11 April 2019
* ETAPS is the primary European forum for academic and industrial researchers working on topics relating to software science. ETAPS, established in 1998, is a confederation of five main annual conferences, accompanied by satellite workshops. ETAPS 2019 is the twenty-second event in the series.
* MAIN CONFERENCES
ESOP: European Symposium on Programming
FASE: Fundamental Approaches to Software Engineering
FoSSaCS: Foundations of Software Science and Computation Structures
POST: Principles of Security and Trust
TACAS: Tools and Algorithms for the Construction and Analysis of Systems
In addition, TACAS ’19 hosts the 8th Competition on Software Verification (SV-COMP).
* INVITED SPEAKERS
Unifying speakers:
Marscha Chechik (University of Toronto, Canada)
Kathleen Fisher (Tufts University, USA)
FoSSaCS invited speaker:
Thomas Colcombet (IRIF, France)
TACAS invited speaker:
Cormac Flanagan (University of California at Santa Cruz, USA)
* IMPORTANT DATES
Abstracts due: 9 November 2018 23:59 AoE
Papers due: 16 November 2018 23:59 AoE
Rebuttal (ESOP, FoSSaCS, POST): 11 - 14 January 2019
Notification: 25 January 2019
Camera-ready versions due: 15 February 2019
* SUBMISSION INSTRUCTIONS
ETAPS conferences solicit contributions of two types: research papers and tool demonstration papers. ESOP and FoSSaCS accept only research papers. FASE, POST and TACAS have multiple types of research papers, see below.
All accepted papers will appear in the proceedings and have presentations during the conference. A condition of submission is that, if the submission is accepted, one of the authors attends the conference to give the presentation.
* OPEN ACCESS
Like ETAPS 2018, the proceedings of ETAPS 2019 will be published in gold open access. The copyright of the papers will remain with the authors. The proceedings will be published in the Advanced Research in Computing and Software Science (ARCoSS) subline of Springer’s Lecture Notes in Computer Science series.
The publisher’s charges for gold open access will be paid by the conference (funded with the participation fees of all
participants). There will be no added cost for authors specifically.

* SATELLITE EVENTS (6-7 April)
A number of satellite workshops will take place before the main conferences: BEHAPI, CREST, DICE-FOPARA, GaLoP, HCVS, HSB, InterAVT, LiVe, MeTRiD, PERR, PLACES, QAPL, SPIoT.

* ORGANIZERS
Jan Kofron and Jan Vitek (general chairs), Barbora Buhnova, Milan Ceska, Ryan Culpepper, Vojtech Horky, Paley Li, Petr Maj, Artem Peletitsyn, David Safranek

HELMUT VEITH STIPEND FOR FEMALE MASTER'S STUDENTS IN COMPUTER SCIENCE
Call for Applications
TU Wien
* Female students in the field of computer science (CS) who plan to pursue (or are currently pursuing) one of the master's programs in Computer Science at the Vienna University of Technology - TU Wien taught in English are invited to apply for the annually awarded Helmut Veith Stipend. The computer science department at Vienna University of Technology - TU Wien, has been ranked among the 70 world's best (THE Times Higher Education Ranking).
* The annually awarded Helmut Veith Stipend for female master students is dedicated to the memory of an outstanding computer scientist who worked in the fields of logic in computer science, computer-aided verification, software engineering, and computer security - Professor Helmut Veith (1971-2016).
* The Helmut Veith Stipend was established with generous support of TU Wien, Wolfgang Pauli Institute and with contributions by family and friends of the late Helmut Veith.
* APPLICATION Applications for funding can be filed before or in parallel with the admissions process. Your application must be submitted electronically to master@logic-cs.at as a single PDF document, by November 30, 2018.

* IMPORTANT DATES
Deadline: November 30, 2018.
* INQUIRIES: Electronically to master@logic-cs.at
* WEBSITE

3rd WORLD CONGRESS ON FORMAL METHODS (FM'19)
Call for Workshop and Tutorial Proposals
Porto, Portugal, October 7-11, 2019
http://formalmethods2019.inesctec.pt/

* FM 2019 is the 23rd international symposium in a series organised by Formal Methods Europe (FME), an independent association whose aim is to stimulate the use of, and research on, formal methods for software development. Every ten years the symposium is organised as a World Congress. For this major event, we are now inviting
proposals for workshops, tutorials, or other satellite events that will complement the main FM Symposium and co-located conferences.

* SUBMISSION INFORMATION

Researchers and practitioners wishing to organise a workshop or tutorial are invited to submit proposals by e-mail to the Workshops and Tutorials Chairs,

Nelma Moreira (nam@dcc.fc.up.pt)
Emil Sekerinski (emil@mcmaster.ca)

For further information please visit
http://formalmethods2019.inesctec.pt/

* IMPORTANT DATES

Submission of proposals: November 16, 2018
Notification of success of proposals: November 23, 2018
Notification of paper acceptance (if applicable): June 14, 2019 (limit date)
FM’19 World Congress: October 7-11, 2019
Workshop/Tutorial dates: October 7-8, 2019 (also October 9-11 if space is an issue)

SPECIAL ISSUE OF THE ARTIFICIAL INTELLIGENCE JOURNAL (AIJ) ON EPISTEMIC PLANNING

* Theme and topics

https://www.journals.elsevier.com/artificial-intelligence/call-for-papers/special-issue-on-epistemic-planning

* Fast publication: Reviewing of submitted articles begins immediately after submission, with first decisions (accept, reject, revisions) made within three months. Accepted articles will be published immediately online on the AIJ website and will also be included in the special issue.

* IMPORTANT DATES

Submission deadline: December 1, 2018
Notification: within 3 months of submission

* SPECIAL ISSUE EDITORS

Vaishak Belle, University of Edinburgh
Thomas Bolander, Technical University of Denmark
Andreas Herzig, CNRS, IRIT Toulouse
Bernhard Nebel, Albert-Ludwigs-Universität Freiburg

16TH EUROPEAN CONFERENCE ON LOGICS IN ARTIFICIAL INTELLIGENCE (JELIA 2019)

First Call for Papers
Rende, Italy, May 8-10, 2019
https://jelia2019.mat.unical.it/

* The aim of JELIA 2019 is to bring together active researchers interested in all aspects concerning the use of logics in Artificial Intelligence to discuss current research, results, problems, and applications of both theoretical and practical nature.

Further information on relevant topics and awards is available at https://jelia2019.mat.unical.it/

* IMPORTANT DATES
Abstract submission deadline 26 November 2018 (23:59 UTC-12)  
Paper submission 03 December 2018 (23:59 UTC-12)  
Notification of acceptance 16 January 2019  
Best paper notification 31 January 2019  
Camera-ready due 28 February 2019  
Online registration opens 01 March 2019  
Conference start 08 May 2019  
* ENQUIRIES  
Please send all enquiries at the email address jelia2019@mat.unical.it  
* GENERAL CHAIR  
Nicola Leone (University of Calabria)  
* PROGRAM CHAIRS  
Francesco Calimeri (University of Calabria)  
Marco Manna (University of Calabria)  

NATURAL LANGUAGE PROCESSING IN ARTIFICIAL INTELLIGENCE (NLPinAI 2019)  
Call for papers  
19 - 21 February, 2019, Prague, Czech Republic  
http://www.icaart.org/NLPinAI.aspx  
* Special Session within the 11th International Conference on Agents and Artificial Intelligence - ICAART 2019  
http://www.icaart.org  
* IMPORTANT DATES:  
Paper Submission: December 20, 2018  
Authors Notification: January 7, 2019  
Camera Ready and Registration: January 15, 2019  
* CHAIRS:  
Roussanka Loukanova Stockholm University, Sweden  
* CONTACT:  
Roussanka Loukanova (rloukanova@gmail.com)  

COMPUTABILITY IN EUROPE 2019 (CiE 2019)  
First Call for Papers  
Durham, United Kingdom  
July 15 - July 19, 2019  
https://community.dur.ac.uk/cie.2019/  
http://www.computability.org.uk  
* IMPORTANT DATES:  
Deadline for article registration (abstract submission): 7 January 2019 AOE  
Deadline for article submission: 14 January 2019 AOE  
Notification of acceptance: 18 March 2019  
Final versions due: 4 April 2019  
Deadline for informal presentations submission: 1 May 2019  
(The notifications of acceptance for informal presentations will be sent a few days after submission.)  
Early registration before: 15 May 2019  
* TUTORIAL SPEAKERS:  
Markus Holzer (JLU Giessen)  
Assia Mahboubi (University of Nantes)  
* INVITED SPEAKERS:
SPECIAL SESSIONS:
- Computational Neuroscience, organised by Noura Al Moubayed (Durham University) and Jason Connolly (Durham University)
- History and Philosophy of Computing, organised by the Council of the HaPoC Commission
- Lowness Notions in Computability, organised by Johanna Franklin (Hofstra University) and Joseph S. Miller (University of Wisconsin-Madison)
- Probabilistic Programming and Higher-Order Computation, organised by Christine Tasson (Paris Diderot University)
- Smoothed and Probabilistic Analysis of Algorithms, organised by Bodo Manthey (University of Twente)
- Transfinite Computations, organised by Sabrina Ouazzani (Paris-Est Creteil University)

PROGRAMME CHAIRS
Daniel Paulusma (Durham University, co-chair)
Giuseppe Primiero (University of Milan, co-chair)

WOMEN IN COMPUTABILITY: We are very happy to announce that within the framework of the Women in Computability programme, we are able to offer some grants of up to 250 EUR for junior female researchers who want to participate in CiE 2019. Applications for this grant should be sent to Liesbeth De Mol, liesbeth.demol@univ-lille3.fr, before 15 May 2019 and include a short cv (at most 2 pages) and contact information for an academic reference. Preference will be given to junior female researchers who are presenting a paper (including informal presentations) at CiE 2019.

ASSOCIATION CIÉ:
http://www.computability.org.uk

CIÉ CONFERENCE SERIES:

FOURTH INTERNATIONAL CONFERENCE ON FORMAL STRUCTURES FOR COMPUTATION AND DEDUCTION (FSCD 2019)
First Call for Papers
24 - 30 June 2019, Dortmund, Germany
http://fscd-conference.org/

IMPORTANT DATES
All deadlines are midnight anywhere-on-earth (AoE); late submissions will not be considered.
Titles and Short Abstracts: 8 February 2019
Full Papers: 11 February 2019
Rebuttal period: 28 March -- 1 April 2019
Authors Notification: 8 April 2019
Final version for proceedings: 22 April 2019

FSCD covers all aspects of formal structures for computation and deduction from theoretical foundations to applications.
two communities, RTA (Rewriting Techniques and Applications) and TLCA (Typed Lambda Calculi and Applications), FSCD embraces their core topics and broadens their scope to closely related areas in logics, models of computation (e.g. quantum computing, probabilistic computing, homotopy type theory), semantics and verification in new challenging areas (e.g. blockchain protocols or deep learning algorithms).

* PROGRAM COMMITTEE CHAIR
  H. Geuvers, Radboud U. Nijmegen

SPRING SCHOOL ON DATABASES, LOGIC AND AUTOMATA (EPIT 2019)
Announcement
April 8-12, 2019, CIRM, Marseille Luminy, France
Webpage of the event: https://conferences.cirm-math.fr/1934.html

* EPIT (ÀLécole de Printemps d’Informatique Théorique, https://epit.irif.fr/) is a French recurrent spring school in theoretical computer science, initiated by Maurice Nivat in 1973. It has since then spanned many exciting topics in foundational computer science, and has become a major event for the research community in France and beyond. The 2019 edition of the EPIT will cover the foundations of data management. It will in particular focus on the fruitful interaction between database theory, logic and automata. A detailed program can be found on the Webpage of the event. In addition, poster sessions will be organised, so that participants who are willing to, will be able to present their work.

* Lectures are intended to be accessible to a wide audience. No prior knowledge of database theory will be assumed, but some familiarity with basic automata theory and logic is recommended. The EPIT 2019 Spring School is primarily addressed to PhD students and young researchers, but more senior participants are also encouraged to join. All courses will be given in English.

* The school will take place at CIRM, the International Center for Mathematical Meetings (https://www.cirm-math.fr/) in Luminy, Marseille. Registration fees, including accommodation and meals at CIRM, will be as moderate as possible (details to be announced soon).

* Pre-registration will soon open on the Webpage of the event.

* Organisers:
  Amelie Gheerbrant: amelie@irif.fr
  Leonid Libkin: libkin@inf.ed.ac.uk
  Luc Segoufin: luc.segoufin@inria.fr
  Pierre Senellart: pierre@senellart.com
  Cristina Sirangelo: cristina@irif.fr

TOPOLOGY, ALGEBRA AND CATEGORIES IN LOGIC (TACL 2019)
Early announcement
Conference: June 17 - 21, 2019 in Nice
School: June 10 - 15, 2019 in Ile de Porquerolles
* Scope: Studying logic via semantics is a well-established and very active branch of mathematical logic with many applications in computer science and elsewhere. The area is characterized by results, tools and techniques stemming from various fields, including universal algebra, topology, category theory, order, and model theory. The programme of the conference TACL 2019 will focus on three interconnecting mathematical themes central to the semantic study of logic and their applications: topological, algebraic, and categorical methods.

* Invited speakers:
  - Samson Abramsky
  - Johan van Benthem
  - Marcel ErnAl'
  - Sam van Gool
  - Wesley Holliday
  - Agi Kurucz
  - Tommaso Moraschini
  - Daniela Petrisan
  - Hilary Priestley
  - Boris Zilber

* Lecturers at the Summer School:
  - Maria Manuel Clementino - Category Theory
  - Andre Joyal - Topos Theory
  - George Metcalfe - Algebraic Methods in Proof Theory
  - Yde Venema - Duality Theory

* Important dates
  - Submission deadline: February 27, 2019
  - Notification to authors: April 10, 2019

Book your hotel as soon as possible. Nice is a popular tourist destination and hotels fill up quickly.

* Detailed information can be found on the webpage.

---

**CATEGORY THEORY 2019 (CT 2019)**

Early announcement
University Of Edinburgh, 7-13 JULY 2019
http://conferences.inf.ed.ac.uk/ct2019/

* We are delighted to announce the Category Theory 2019 conference at University Of Edinburgh between 7-13 July 2019.
* Details will follow by email and on the conference website:
  http://conferences.inf.ed.ac.uk/ct2019/

---

**INTERNATIONAL CONFERENCE ON RELIABILITY, SAFETY AND SECURITY OF RAILWAY SYSTEMS: MODELLING, ANALYSIS, VERIFICATION AND CERTIFICATION (RSSRAIL 2019)**

Call for Papers
June 4-6, 2019, Lille, France
https://conferences.ncl.ac.uk/rssrail2019/

* The conference aims to bring together researchers and engineers interested in building critical railway applications and
systems. This will be a working conference in which research challenges and progress will be discussed and evaluated by both researchers and engineers, focusing on their potential to be deployed in industrial settings.

* Topics of particular interest include: Safety in development processes and safety management, Combined approaches to safety and security, System and software safety analysis, Formal modelling and verification techniques, System reliability, Validation according to the standards, Safety and security argumentation, Fault and intrusion modelling and analysis, Evaluation of system capacity, energy consumption, cost and their interplay, Tool and model integration, tool chains, Domain-specific languages and modelling frameworks, Model reuse for reliability, safety and security, Modelling for maintenance strategy engineering.

* IMPORTANT DATES
  Abstract submission deadline: January 5, 2019

23RD INTERNATIONAL SYMPOSIUM ON FORMAL METHODS - 3RD WORLD CONGRESS ON FORMAL METHODS (FM 2019)
First Call for Papers
Porto, Portugal, October 7-11, 2019
http://formalmethods2019.inesctec.pt/

* FM 2019 is the 23rd international symposium in a series organised by Formal Methods Europe (FME), an independent association whose aim is to stimulate the use of, and research on, formal methods for software development. Every 10 years the symposium is organised as a World Congress. Twenty years after FM 1999 in Toulouse, and 10 years after FM 2009 in Eindhoven, FM 2019 is the 3rd World Congress on Formal Methods. This is reflected in a PC with members from over 40 countries. Thus, FM 2019 will be both an occasion to celebrate and a platform for enthusiastic researchers and practitioners from a diversity of backgrounds to exchange their ideas and share their experience.

* IMPORTANT DATES
  Abstract submission: 28 March, 2019
  Full paper submission: 11 April, 2019, 23:59 AoE
  Notification: 11 June, 2019
  Camera ready: 9 July, 2019
  Conference: 7-11 October, 2019

* Topics of Interest
FM 2019 encourages submissions on formal methods in a wide range of domains including software, computer-based systems, systems-of-systems, cyber-physical systems, human-computer interaction, manufacturing, sustainability, energy, transport, smart cities, and healthcare. We particularly welcome papers on techniques, tools and experiences in interdisciplinary settings. We also welcome papers on experiences of formal methods in industry, and on the design and validation of formal methods tools. The broad topics of interest for FM 2019 include, but are not limited to:

- Interdisciplinary formal methods: Techniques, tools and experiences demonstrating the use of formal methods in...
interdisciplinary settings.
- Formal methods in practice: Industrial applications of formal methods, experience with formal methods in industry, tool usage reports, experiments with challenge problems. The authors are encouraged to explain how formal methods overcame problems, led to improved designs, or provided new insights.
- Tools for formal methods: Advances in automated verification, model checking, and testing with formal methods, tools integration, environments for formal methods, and experimental validation of tools. The authors are encouraged to demonstrate empirically that the new tool or environment advances the state of the art.
- Formal methods in software and systems engineering: Development processes with formal methods, usage guidelines for formal methods, and method integration. The authors are encouraged to evaluate process innovations with respect to qualitative or quantitative improvements. Empirical studies and evaluations are also solicited.
- Theoretical foundations of formal methods: All aspects of theory related to specification, verification, refinement, and static and dynamic analysis. The authors are encouraged to explain how their results contribute to the solution of practical problems with formal methods or tools.
* Best Paper Award: At the conference, the PC Chairs will present an award to the authors of the submission selected as the FM 2019 Best Paper.
* General Chair
JosÃ© Nuno Oliveira, INESC TEC & University of Minho, PT
* Program Committee Chairs
Maurice ter Beek, ISTI-CNR, Pisa, IT
Annabelle McIver, Macquarie University, AU

PHD STUDENT POSITION IN DATABASE THEORY AND LOGIC
* Database Theory and Logic
* joint between LaBRI (Bordeaux, FR) and IRIF (Paris, FR)
* 3 years
* Link: https://quid.labri.fr/documents/phd.html
* The IRIF lab in Paris and the LABRI lab in Bordeaux, France have funding for a co-supervised PhD studentship in database theory starting in 2019. The PhD topic is in the area of foundations of data management, focusing on querying inconsistent data. This PhD topic is part of a larger projet QUID (Efficient Querying for Incomplete and Inconsistent Data), funded by the French research agency ANR. The project involves researchers from two other research labs in France: Ecole Normale Superieure (Paris) and the Institut Gaspard Monge (Marne-la-Vallee). Candidates should have a strong background in theoretical computer science, preferably in automata, logic, verification, or finite model theory. Some prior knowledge of database theory and systems is also a plus.
* Contact:
Cristina Sirangelo - cristina@irif.fr
AI*IA INCOMING AND OUTGOING MOBILITY GRANTS 2018 - CALL FOR RESEARCH VISITS
* To favour mobility of young researchers the Italian Association for Artificial Intelligence (AI*IA) issues the AIxIA Incoming Mobility Grants for 2018. Applications are solicited for funding a research visit of a PhD student enrolled at a foreign University to an Italian institution and for a research visit abroad of a PhD student enrolled at an Italian University. The central aim of these long visits is to build a research bridge between researchers and to create a solid basis for long term collaborations. Moreover, the visit has to lead to a submission of an article on a joint research topics to the Intelligenza Artificiale journal (www.iospress.nl/journal/intelligenza-artificiale/).
* For the incoming call eligibility for the visiting student is to be enrolled full-time in a PhD programme at a foreign University. Funding is available for 2 students.
* For the outgoing call applications can be made by students enrolled full-time in a PhD programme at an Italian University. The applicant must be a member of the Association for 2018. If she/he is not a member for 2018 she/he must register before applying.
* The visits should start between the 1st of January 2019 and the 31th of December 2019.
* Deadline for applications: November 30th, 2018
  Notification of grants: December 15th, 2018
* Further information:
  https://groups.google.com/a/aixia.it/forum/#!msg/aixia/VHeiyHXKocY/jZEjryGPAGAJ
  The applications must be sent by email to incoming@aixia.it. The applications will be examined by a committee composed by members of the AI*IA Board of Directors.

POSTDOCTORAL RESEARCHER IN LOGIC, MILAN
Project: Logical Foundations and Applications of Depth-Bounded Probability
Duration: 2 years
Deadline for application: 9 November 2018
http://www.unimi.it/ricerca/assegni_ricerca/123679.htm
* Applicants are advised to contact the PIs of the project, Marcello D’Agostino <marcello.dagostino@unimi.it> or Hykel Hosni <hykel.hosni@unimi.it>, for further information about formal requirements.
* We are looking for a very strong and highly motivated postdoctoral researcher in Logic to join Marcello D’Agostino and Hykel Hosni who are the PIs of the project “Logical Foundations and Applications of Depth-Bounded Probability”. This project is part of a 5 years “Excellence Scheme” which has been awarded in 2017 to The Department of Philosophy at the University of Milan “La Statale” in recognition of its leading role in research and innovative teaching.
* Further information:
  http://www.unimi.it/ricerca/assegni_ricerca/123679.htm
The Special Interest Group on Logic and Computation is the premier international community for the advancement of logic and computation, and formal methods in computer science, broadly defined.

The Association for Computing Machinery (ACM) is an educational and scientific computing society which works to advance computing as a science and a profession. Benefits include subscriptions to Communications of the ACM, MemberNet, TechNews and CareerNews, full and unlimited access to online courses and books, discounts on conferences and the option to subscribe to the ACM Digital Library.

- SIGLOG (ACM Member) .................................................. $ 25
- SIGLOG (ACM Student Member & Non-ACM Student Member) .................................. $ 15
- SIGLOG (Non-ACM Member) .................................................. $ 25
- ACM Professional Membership ($99) & SIGLOG ($25) .................................................. $124
- ACM Professional Membership ($99) & SIGLOG ($25) & ACM Digital Library ($99) ...................... $223
- ACM Student Membership ($19) & SIGLOG ($15) .................................................. $ 34

payment information

Name ____________________________
ACM Member # ____________________
Mailing Address ____________________
__________________________________
__________________________________
City/State/Province ________________
ZIP/Postal Code/Country ____________
Email ______________________________
Mobile Phone _______________________
Fax _______________________________

Credit Card Type:  □ AMEX □ VISA □ MC
Credit Card #: ______________________
Exp. Date __________________________
Signature __________________________

Mailing List Restriction
ACM occasionally makes its mailing list available to computer-related organizations, educational institutions and sister societies. All email addresses remain strictly confidential. Check one of the following if you wish to restrict the use of your name:

- □ ACM announcements only
- □ ACM and other sister society announcements
- □ ACM subscription and renewal notices only

Questions? Contact:
ACM Headquarters
2 Penn Plaza, Suite 701
New York, NY 10121-0701
voice: 212-626-0500
fax: 212-944-1318
email: acmhelp@acm.org

Remit to:
ACM
General Post Office
P.O. Box 30777
New York, NY 10087-0777

SIGAPP

www.acm.org/joinsigs

Advancing Computing as a Science & Profession