# Table of Contents

## General Information

1. From the Editor  *Andrzej Murawski*
2. Chair’s Letter  *Prakash Panangaden*

## Technical Columns

3. Automata  *Mikołaj Bojańczyk*
18. Complexity  *Neil Immerman*
45. Security and Privacy  *Matteo Maffei*
70. Semantics  *Michael Mislove*

## Announcements

85. SIGLOG Monthly 172
# 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>Natarajan Shankar</td>
<td>SRI International</td>
</tr>
<tr>
<td>Secretary</td>
<td>Alexandra Silva</td>
<td>Radboud University Nijmegen</td>
</tr>
<tr>
<td>EACSL President</td>
<td>Anuj Dawar</td>
<td>University of Cambridge</td>
</tr>
<tr>
<td>EATCS President</td>
<td>Luca Aceto</td>
<td>Reykjavik University</td>
</tr>
<tr>
<td>ACM ToCL E-in-C</td>
<td>Dale Miller</td>
<td>INRIA and LIX, École Polytechnique</td>
</tr>
<tr>
<td></td>
<td>Andrzej Murawski</td>
<td>University of Warwick</td>
</tr>
<tr>
<td></td>
<td>Véronique Cortier</td>
<td>CNRS and LORIA, Nancy</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>

## COLUMN EDITORS

<table>
<thead>
<tr>
<th>Area</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>CISPA, Saarland University</td>
</tr>
<tr>
<td>Semantics</td>
<td>Mike Mislove</td>
<td>Tulane University</td>
</tr>
<tr>
<td>Verification</td>
<td>Neha Rungta</td>
<td>SGT Inc. and NASA Ames</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

With this issue, SIGLOG News is entering its second year of existence. Many thanks to all the authors for their contributions and to the column editors for attracting excellent submissions!

In this issue

– In Mikołaj Bojańczyk's Automata Column, Thomas Place and Marc Zeitoun write about the quantifier alternation hierarchy and first-order logic over words.
– In Neil Immerman's column on Complexity, Jakob Nordström surveys connections between proof complexity and SAT solving.
– Konstantinos Chatzikokolakis, Catuscia Palamidessi and Marco Stronati report on geo-indistinguishability in Matteo Maffei's column on Security and Privacy.
– Prakash Panangaden discusses probabilistic bisimulation in Michael Mislove's column on Semantics.
– And, finally, there are numerous calls for papers and participation to look through in our monthly Bulletin.

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

Enjoy!

Andrzej Murawski
University of Warwick
SIGLOG News Editor
We are up to 250 members! That is not bad and more than I had projected originally. I hope that the upcoming Summer conferences will give a boost to our numbers. As I said in my last message we will be having two joint events this year: Mathematical Foundations of Programming Semantics will be held jointly with the Sixth Conference on Algebra and Coalgebra in Computer Science. Both are to be held in Nijmegen during the week of June 22nd. Later on LICS and ICALP will be co-located in Kyoto. It is hard to believe that both MFPS and LICS are 31 and 30 years old respectively!

I was there at the first LICS in Boston where much of the excitement was about type theory and especially polymorphism and the new models of polymorphic \(\lambda\)-calculus that had been discovered by Moggi. There was also a lot of interest in the then new Calculus of Constructions which is alive and well today in Coq. As well the computer science community was introduced to linear logic for the first time. It is amazing how enduring these ideas have proved. Let us see what emerges this summer that we will be talking about in 2045.

Prakash Panangaden  
McGill University  
ACM SIGLOG Chair
The foundational result of automata and logic is that regular languages of finite words are exactly those which can be defined in monadic second-order logic. For example, the language of words of even length is defined by a formula which says there exists a set $X$ of positions that contains the first position but not the last one, and furthermore for each position $x$, the set $X$ contains either $x$ or its successor, but not both.

A natural question is: what about first-order logic? Which languages of finite words can one define when one can only quantify over positions and not sets of positions? It is not difficult to show that first-order logic is strictly less expressive than monadic second-order logic. For example, the abovementioned language of words of even length cannot be defined in first-order logic. A beautiful result, which was proved by Schützenberger, McNaughton and Papert, says that the languages definable in first-order logic are exactly those that can be recognised by homomorphisms into finite monoids that do not contain any group. (As opposed to monadic second-order logic, which corresponds to languages that are recognised by homomorphisms into arbitrary finite monoids, possibly including groups.)

The above result explained in depth the expressive power of first-order logic on finite words. What about fragments of first-order logic, e.g. what about formulas with restricted quantifier alternation? People have asked: which regular languages are definable by formulas with an $\exists^*$ quantifier prefix?, or maybe by Boolean combinations of such formulas?, or maybe by formulas with an $\exists^*\forall$ quantifier prefix?, or maybe by Boolean combination of such formulas?, etc. So far, this study has covered the first five or six levels of the quantifier alternation hierarchy, and what is maybe surprising, this study is actually worth the effort. Each new level is quite interesting and original with respect to the previous ones. This column reports on the quest to understand first-order logic for finite words, and we are fortunate enough to hear the story from authors of the most recent breakthroughs in the field, Thomas Place and Marc Zeitoun.
In this survey, we present ideas developed until recently in order to understand the expressive power of logical fragments in the quantifier alternation hierarchy of first-order logic interpreted on finite words.

1. INTRODUCTION

This paper surveys milestones, from early to recent results, on the expressiveness of fragments of first order logic interpreted on finite words. In this context, “understanding the expressive power of a fragment of first-order logic $F$” often amounts to finding an algorithm for the following decision problem:

**Input** A regular language of finite words $L$.

**Question** Can $L$ be defined by a sentence of $F$?

In other words, we ask whether the input language belongs to the class of languages defined by the logic, hence the name of the problem: the $F$-membership problem. Having an $F$-membership algorithm in hand amounts to having an effective description of all regular properties that $F$ can express. This is why obtaining a membership algorithm is viewed as the goal to strive for when trying to get a precise understanding of a logic. This problem is always difficult as it is a semantic question: whether a regular language is definable in $F$ may not be apparent in the syntax that defines it.

Our main objective is to outline advances that have led to membership algorithms for logical fragments in a well-known and natural hierarchy of first order definable languages, the quantifier alternation hierarchy. Intuitively, the notion of quantifier alternation classifies first order logic according to the difficulty we have to define languages. To obtain such a classification, we need a notion of hardness, i.e., a good measure telling how complex a property is. Quantifier alternation is a natural such measure: a language is considered complicated when we need many switches between blocks of $\exists$ quantifiers and blocks of $\forall$ quantifiers to express it. In mathematics, usually few such alternations are used (one quickly gets lost beyond 4 or 5). This motivates the study of what can be expressed with a fixed number of quantifier alternations, and, already importantly, with few of them. Observe that the notion of quantifier alternation is, again, semantic and not syntactic: given a first order definable language, we look at the minimum number of quantifier alternations that is needed to define the language.
The search for membership algorithms for each level in this hierarchy started in 1965 when an algorithm was found for full first-order logic [Schützenberger 1965; McNaughton and Papert 1971]. This investigation is still ongoing work today, with results as recent as 2015 [Place 2015; Almeida et al. 2015]. The main developments were separated by decades. This is explained by the fact that each of them required to add new conceptual ingredients to the mix. In this paper, we survey the story of this quest by organizing it around three milestones, corresponding to the three main concepts that are needed to explain the most recent results.

2. LOGICAL FRAGMENTS OF FIRST ORDER LOGIC

For the whole paper, we assume that an arbitrary alphabet $A$ is fixed and that we consider finite words over this alphabet. As usual, the set of all these words is denoted by $A^*$. A language is simply a set of words.

One can view a word as a logical structure made of an ordered sequence positions carrying labels in $A$. For example the word $abaac$ is made of positions $0 < 1 < 2 < 3 < 4$ labeled by $a, b, a, a$ and $c$, respectively. In first-order logic over words ($\text{FO}(\prec)$), one can quantify over positions in a word and use the following predicates to test properties of these positions:

- for each $a \in A$, a unary predicate $P_a$ that selects positions labeled with an $a$.
- a binary predicate `$\prec$', which is interpreted as the linear order over the positions.

Moreover, as usual, one is allowed to use boolean connectives within sentences. Each sentence of $\text{FO}(\prec)$ defines a language: the language of all words that satisfy the sentence. For instance, the sentence `$\exists x (P_a(x) \land \exists y x < y)$" defines the language of all words in which there exists a position which carries an $a$ and is not the rightmost one. Hence, $\text{FO}(\prec)$ defines a class of languages: the class of all languages that can be defined using an $\text{FO}(\prec)$ sentence. We also denote this class by $\text{FO}(\prec)$. In particular, $\text{FO}(\prec)$ is a subclass of that of all regular languages. This follows from the well-known theorem of Büchi, Elgot and Trakhtenbrot, which states that being regular is equivalent to being definable in the more expressive monadic second order logic (MSO) [Büchi 1960; Elgot 1961; Trakhtenbrot 1961].

The notion of quantifier alternation is a natural way to stratify the class $\text{FO}(\prec)$ as a hierarchy. It classifies first-order sentences by counting the number of alternations between existential and universal quantifiers inside their prenex normal form. An important remark is that one can find two different “quantifier alternation hierarchies of first-order logic" in the literature. They correspond to two different (but equivalent) ways of defining first-order logic over words. We start with the hierarchy corresponding to our definition, which we call the order hierarchy.

**Order Hierarchy.** As explained, one can classify first-order sentences by counting the number of alternations between existential and universal quantifiers inside their prenex normal form. An important remark is that one can find two different “quantifier alternation hierarchies of first-order logic" in the literature. They correspond to two different (but equivalent) ways of defining first-order logic over words. We start with the hierarchy corresponding to our definition, which we call the order hierarchy.

Given $n \geq 1$, an $\text{FO}(\prec)$ sentence is said to be $\Sigma_n(\prec)$ (resp. $\Pi_n(\prec)$) if its prenex normal form has $n$ blocs of nested quantifiers (or, equivalently, $(n-1)$ alternations) and starts with an $\exists$ (resp. $\forall$) quantifier. That is, a $\Sigma_n(\prec)$ sentence is a sentence whose prenex normal form has the following shape:

$$\exists^n \forall^n \exists^n \cdots \varphi$$

where $\varphi$ is a quantifier-free $\text{FO}(\prec)$ formula. It is straightforward to see that the classes of languages corresponding to $\Sigma_n(\prec)$ and $\Pi_n(\prec)$ are closed under union and intersection. On the other hand, these classes are not closed under complement (the negation of
A $\Sigma_n(<)$ sentence is a $\Pi_n(<)$ sentence, and conversely. This is why one also considers the $\mathcal{B}\Sigma_n(<)$ sentences, which are boolean combinations of $\Sigma_n(<)$ and $\Pi_n(<)$ formulas.

Clearly, we have $\Sigma_n(<) \subset \mathcal{B}\Sigma_n(<) \subset \Sigma_{n+1}(<)$. The first natural question is whether these inclusions are strict, and it turns out that this is the case. The hierarchy is depicted in Figure 1, where the color indicates the status of the class with respect to the membership problem.

**Enriched Hierarchy.** In $\text{FO}(<)$, several natural predicates can be defined from the linear order “$<$”. This is the case for the predicate testing that a position is the leftmost one in the word, by the formula $\text{min}(x) := \neg \exists y \ y < x$, or symmetrically the rightmost one by a $\text{max}(x)$ formula. Another natural predicate is the successor, which tests whether two positions are consecutive. It is defined by $\text{+1}(x, y) := (x < y) \land \neg \exists z (x < z \land z < y)$.

It follows that one can present an alternate definition of first-order logic over words, $\text{FO}(<, \text{+1})$, in which these predicates are explicitly allowed in the signature. While the two definitions are equivalent for full first-order logic, this is not the case for classes of the quantifier alternation hierarchy. Indeed, replacing the predicates $\text{min}(x)$, $\text{max}(x)$ or $\text{+1}(x, y)$ by the formulas above inside a sentence may increase its quantifier alternation. It can be shown that this is actually unavoidable. Thus, we get a second quantifier alternation hierarchy. We call it the *enriched hierarchy* and denote its classes by $\Sigma_n(<, \text{+1})$, $\Pi_n(<, \text{+1})$ and $\mathcal{B}\Sigma_n(<, \text{+1})$.

As we will see, the investigation of the membership problem for the two hierarchies is strongly related. In fact, the state of the art is the same for both, as depicted in Figure 1.

### 3. FIRST MILESTONE: SCHÜTZENBERGER’S THEOREM

We begin our survey with the milestone that started it all: Schützenberger’s Theorem [Schützenberger 1965]. Naturally, before fragments in the quantifier alternation hierarchy of first-order logic were looked at, first-order logic was investigated as a whole. The main result in this investigation is Schützenberger’s Theorem from which...
a membership algorithm for first-order logic can be easily obtained. This theorem was historically the first one and it defined a template that is still followed nowadays. Before we present the theorem and explain the relevance of the approach taken by Schützenberger, let us make an important remark.

In its original statement, Schützenberger’s Theorem does not actually refer to first-order logic: it is about the class of star-free languages. This is the smallest class of languages containing all finite languages and closed under boolean operations (including complement) and concatenation product. On the other hand, the class is not closed under the Kleene star, hence the name “star-free”. Therefore, originally, what followed from the theorem was a membership algorithm for the class of star-free languages. The connection with first-order logic was made later by McNaughton and Papert [McNaughton and Papert 1971] who proved that the classes of first-order definable languages and star-free languages are the same.

An important point is that while the membership algorithm for \( \text{FO}(\prec) \) is the consequence of two results, these two results are not of the same importance: the core of the argument is Schützenberger’s proof. Indeed, that “\( \text{FO}(\prec) = \text{Star-Free} \)” is proved via direct rewriting of formulas (essentially, this amounts to proving that concatenation can be simulated with existential quantification, and conversely). In other words, the proof is mainly syntactic: if we have an \( \text{FO}(\prec) \) formula \( \varphi \) that defines a language \( L \) in hand, we do not have to analyze \( L \), we can directly rewrite \( \varphi \) into a star-free description of \( L \). On the other hand, proving Schützenberger’s Theorem (i.e., deciding whether a regular language \( L \) is \( \text{FO}(\prec) \)-definable or star-free) requires a semantic understanding of the input language \( L \). In particular, given a regular language \( L \) that is first-order definable, one cannot directly compute an \( \text{FO}(\prec) \) formula defining \( L \) out of any representation of \( L \) (its syntactic monoid) and then, to rebuild a first-order formula from this abstract representation.

This last remark underlines the fact that Schützenberger’s Theorem is stated and proved within a general framework. This framework is tailored to the membership problem and is generic in the sense that it can be applied to many classes of languages besides \( \text{FO}(\prec) \) itself. We start by presenting this framework and by explaining why it is that relevant for the membership problem.

### 3.1. A Framework for the Membership Problem: the Syntactic Approach

Our goal is now to define a framework that is suitable for investigating the membership problem associated to some class of languages \( \mathcal{C} \). Recall that in the membership problem, we want to know whether a single input regular language is definable in \( \mathcal{C} \). In order to understand Schützenberger’s approach to this problem, one has to ask the following question: what is the advantage of having a regular input? The answer can be found in the following result, which is the main ingredient of Schützenberger’s approach.

**Main Ingredient: Myhill-Nerode Theorem.** Myhill-Nerode Theorem is a well-known characterization of regular languages, as those whose syntactic congruence has finite index. To any language \( L \subseteq A^* \) (not only regular ones), one can associate an equivalence relation \( \sim_L \) on the set \( A^* \): the syntactic congruence of \( L \). Intuitively, two words are equivalent if they cannot be distinguished by \( L \), even when embedded in the same context. Formally, the definition is as follows:

\[
u \sim_L v \quad \text{if and only if} \quad \forall x, y \in A^* \quad xuy \in L \iff xvy \in L.
\]

By definition, \( L \) is a union of equivalence classes of \( \sim_L \). In other words, the syntactic congruence of \( L \) breaks down \( L \) in fundamental simpler parts. Furthermore, the set of these fundamental parts has a structure. Indeed, as the name suggests, it is simple to
prove that the syntactic congruence is a congruence for the concatenation operation: if \( u \sim_L v \) and \( u' \sim_L v' \), then \( uu' \sim_L vv' \). This has two consequences:

1. The set of equivalence classes of \( \sim_L \) is a monoid when equipped with the concatenation operation: if \( K, K' \) are equivalence classes, then \( KK' \) is included in an equivalence class. This monoid, denoted by \( M_L \), is called the syntactic monoid of \( L \).

2. The map \( \alpha_L : A^* \to M_L \) that associates its equivalence class to each word is a monoid morphism (for any \( u, v \in A^* \), \( \alpha_L(uv) = \alpha_L(u)\alpha_L(v) \)). It is called the syntactic morphism of \( L \).

The syntactic morphism \( \alpha_L \) defines several languages: all languages which are unions of equivalence classes of \( \sim_L \). We speak of the set of languages recognized by \( \alpha_L \). In particular, observe that both \( L \) and its complement are among these recognized languages. From Myhill-Nerode Theorem, we know that the syntactic monoid and the syntactic morphism of a language \( L \) become very relevant objects when \( L \) is regular.

**Theorem 3.1 (Myhill-Nerode Theorem).** Let \( L \) be a language. Then \( L \) is regular if and only if \( \sim_L \) has finite index, i.e., if and only if the syntactic monoid of \( L \) is finite.

Note that when \( M_L \) is finite, \( \alpha_L \) can be finitely represented: as a morphism it is defined by its restriction to letters of the alphabet. Therefore, what Myhill-Nerode Theorem gives us is a finite canonical representation of any regular language: its syntactic morphism \( \alpha_L : A^* \to M_L \). Moreover, the proof of the theorem is constructive: from any representation of a regular language \( L \), one can compute \( \alpha_L \).

The main point of Schützenberger’s approach to the membership problem is that the syntactic morphism is the “right” representation of a regular language when trying to decide semantic properties. In particular, it follows from Theorem 3.2 below that this representation is tailored to the investigation of the \( \mathcal{C} \)-membership problem for classes of languages \( \mathcal{C} \) that satisfy the following conditions:

1. \( \mathcal{C} \) is nonempty and closed under boolean operations (including complement).
2. \( \mathcal{C} \) is closed under right and left quotients: for any \( w \in A^* \) and \( L \in \mathcal{C} \)

\[ w^{-1}L = \{u \mid uw \in L\} \in \mathcal{C} \quad \text{and} \quad LW^{-1} = \{u \mid uw \in L\} \in \mathcal{C}. \]

Note that it is well-known and simple to verify that \( \text{FO}(\prec) \) itself fulfills these two conditions.

**Theorem 3.2.** Let \( \mathcal{C} \) be a nonempty class of regular languages that is closed under boolean operations and quotients. Then, for any regular language \( L \), \( L \) belongs to \( \mathcal{C} \) if and only if every language recognized by \( \alpha_L \) belongs to \( \mathcal{C} \).

The statement of Theorem 3.2 might seem surprising as it is not entirely obvious that we gain something from it: we reduced the problem of deciding whether a single regular language \( L \) belongs to \( \mathcal{C} \) to the problem of deciding whether several regular languages belong to \( \mathcal{C} \). However, these languages are not any languages: combined, they form a finitely presentable piece of syntax that defines the input language \( L \). In view of this, what we really obtain from Theorem 3.2 is that deciding a semantic property of \( L \) (whether \( L \) belongs to \( \mathcal{C} \)) can be reduced to deciding a syntactic property of the syntactic morphism \( \alpha_L \). Usually, such a result is called a decidable characterization of \( \mathcal{C} \): one proves that membership of a regular language \( L \) in \( \mathcal{C} \) is equivalent to an easily decidable syntactic property of its syntactic morphism \( \alpha_L \).

Naturally, the general approach that we have presented so far tells nothing of what this syntactic property should be, as it is specific to \( \mathcal{C} \). This is where the real investigation on the class \( \mathcal{C} \) takes place: finding the syntactic property and proving that it...
is equivalent to definability in $C$ remains a hard problem for most classes $C$. In other words, the syntactic morphism is a suitable and convenient framework for investigating the membership problem associated to $C$:

1. it provides an elegant way to state the algorithm and,
2. it provides convenient tools and properties to prove its correction (we shall detail this point below).

On the other hand, what the syntactic morphism is not is a generic solution to the membership problem for all classes $C$. The best (and historically first) example of this is Schützenberger’s Theorem itself, which we now state.

**Theorem 3.3 (Schützenberger-McNaughton-Papert).** Let $L$ be a regular language. Then, the three following conditions are equivalent:

- $L$ is definable in $\text{FO}(\prec)$.
- $L$ is star-free.
- $M_L$ is aperiodic: for every $s \in M_L$, we have
  \[ s^\omega = s^{\omega+1}. \] (AP)

The symbol $\omega$ in (AP) denotes a natural number that can be computed from $M_L$, which satisfies the following property: for any $s \in M$, $s^\omega$ is an idempotent ($s^\omega = s^\omega s^\omega$). It follows that the last item in Theorem 3.3 is a syntactic condition which can easily be decided for $M_L$. Therefore, Theorem 3.3 yields the desired membership algorithm.

As we shall see in the next subsection, Theorem 3.3 is not an isolated case: many classes $C$ were given decidable characterizations with theorems having an elegant statement resembling that of Theorem 3.3. This illustrates Item (1) above: the syntactic morphism provides an elegant way to state membership algorithms. However, the relevance of the syntactic approach of Schützenberger’s Theorem does not stop at the statement of the result: as claimed in Item (2) above, the syntactic morphism also provides the right tools to prove such theorems. Let us explain why in more detail.

In order to prove a statement similar to Theorem 3.3 for some class $C$ (such as $\text{FO}(\prec)$ in the case of Theorem 3.3), one has to prove that for any regular language $L$, we have $L \in C$ if and only if $\alpha_L$ satisfies some syntactic property (such as $M_L$ is aperiodic in the case of Theorem 3.3). It turns out that for all known cases, including $\text{FO}(\prec)$, the difficult direction is the “if” one: starting from an abstract representation of $L$ satisfying an abstract property, one has to build a formula that defines $L$. In particular, this makes the proof of this direction very interesting: it yields a canonical way to construct a formula for any language of $C$.

By Theorem 3.2, what one needs to prove is that any language recognized by $\alpha_L$ is definable in $\text{FO}(\prec)$. Very roughly, this is achieved by decomposing each recognized language as the composition (by concatenation, union, intersection or complement) of simpler languages (i.e., recognized by a simpler syntactic morphism to which induction can be applied). This is where the algebraic structure of the syntactic morphism is useful: it may be used to define induction parameters and find clever ways to make these decompositions.

### 3.2. After Schützenberger: First Classes of the Hierarchies

Schützenberger’s approach served as a template that was applied (and is still applied) to many natural classes of languages. In particular, since Schützenberger’s Theorem was about star-free languages, researchers started considering hierarchies of star-free languages (which would later be proved to correspond to quantifier alternation hierarchies within $\text{FO}(\prec)$).
The Enriched Hierarchy. The first hierarchy to be considered was the Dot-Depth Hierarchy of Brzozowski and Cohen [Brzozowski and Cohen 1971] in which each level counts the minimal number of nested alternations between concatenation and complement operations that are needed to build a star-free language. This hierarchy was proved to be strict (each level is strictly larger than the previous one) in [Brzozowski and Knast 1978]. Note that the link between this classification method and quantifier alternation is quite intuitive: since concatenation corresponds to existential quantification, alternating complement and concatenation is connected to alternating quantifiers. The formal connection was made later by Thomas [Thomas 1982], who proved that the dot-depth hierarchy corresponds to the enriched quantifier hierarchy of first-order logic: the languages of dot-depth $i$ are the languages definable in $\mathcal{B}\Sigma_i(<, +1)$.

Remark 3.4. Observe that we did not mention the logics $\Sigma_i(<, +1)$ and $\Pi_i(<, +1)$. This is because the original definition of the dot-depth hierarchy by Brzozowski and Cohen did not include them. They were only added later as “half-levels”: $\Sigma_i(<, +1)$ corresponds to languages of dot-depth $i - \frac{1}{2}$. We will detail this point in Section 4, as treating these classes requires to generalize the syntactic approach.

Despite its early definition (1971), it was not until 1983 that a membership algorithm for the first level of the dot-depth hierarchy was found (i.e., the level corresponding to $\mathcal{B}\Sigma_1(<, +1)$). This result, due to Knast [Knast 1983], was presented and proved with the syntactic approach: it states that a language is definable in $\mathcal{B}\Sigma_1(<, +1)$ if and only its syntactic morphism satisfies an easy to decide syntactic property, which can be stated as an equation similar to (AP).

Despite this initial success, level 2 of the dot-depth hierarchy (i.e., $\mathcal{B}\Sigma_2(<, +1)$) was not given a membership algorithm until 2014 [Place and Zeitoun 2014a]. In fact, after Knast’s result, the focus was quickly shifted to the second hierarchy, the order hierarchy, following a result by Straubing [Straubing 1985] suggesting that investigating $\mathcal{B}\Sigma_2(<)$ was the right approach to solving membership for $\mathcal{B}\Sigma_2(<, +1)$.

The Order Hierarchy. Surprisingly, the simpler order quantifier alternation hierarchy was not considered until much later than the dot-depth hierarchy. It was first introduced independently by Thérien [Thérien 1981] and Straubing [Straubing 1981] as another hierarchy of star-free languages: the Straubing-Thérien hierarchy. It was then observed by Perrin and Pin [Perrin and Pin 1986] that it actually corresponds to the order hierarchy.

It was proved by Straubing [Straubing 1985] (for the integer $\mathcal{B}\Sigma_i$ levels) and then by Pin and Weil [Pin and Weil 2002] (for the half $\Sigma_i$ levels) that this hierarchy is in a sense the most fundamental of the two. More precisely, they proved that for $i \geq 2$, the membership problem for $\mathcal{B}\Sigma_i(<, +1)$ (resp. $\Sigma_i(<, +1)$) can be effectively reduced to the same problem for $\mathcal{B}\Sigma_2(<)$ (resp. $\Sigma_i(<)$). This explains why, after 1985, efforts have mostly aimed at obtaining membership algorithms for levels of the order hierarchy (in fact the result of [Place and Zeitoun 2014a] cited above for $\mathcal{B}\Sigma_2(<, +1)$ is actually a membership algorithm for $\mathcal{B}\Sigma_2(<)$).

Ironically, the first level of the Straubing-Thérien hierarchy corresponding to $\mathcal{B}\Sigma_1(<)$ was given a membership algorithm before the actual hierarchy was even defined. Indeed, $\mathcal{B}\Sigma_1(<)$ corresponds to the class of piecewise testable languages, which was first investigated independently from the hierarchy and given a decidable characterization by Simon in [Simon 1975]. This result is usually referred to as Simon’s Theorem (not to be confused with Simon’s Factorization Forests Theorem [Simon 1990]) and is arguably the second most famous result of this kind, after Schützenberger’s Theorem.

Many attempts at generalizing these results to $\mathcal{B}\Sigma_2(<)$ were made over the years (see [Pin 2011; Almeida and Klima 2010; Pin 1998; Pin and Straubing 1985] for example).
However, until 2014, the only result that was known was partial, working only when
the alphabet has size $2$ [Straubing 1988]. This can be explained by the fact that the
algorithm that was finally obtained in [Place and Zeitoun 2014a] relies on two additional
ingredients that fall outside of the syntactic approach that we presented thus far:

1. Investigating the “half-levels” $\Sigma_i(\prec)$.
2. Considering a problem that is more general than membership for these “half-levels”:
   the separation problem.

The two following sections are devoted to the presentation of these two ingredients.

4. SECOND MILESTONE: CLASSES THAT ARE NOT CLOSED UNDER COMPLEMENT

An issue with the general approach that we outlined in the previous section is that it
can only be applied to classes of languages that are closed under complement. Given a
regular language $L$, the set of languages that are recognized by its syntactic morphism
contains both $L$ and its complement. Therefore, a class of languages $C$ that satisfies
Theorem 3.2 (which is the basis of the syntactic approach) must be closed under
complement.

This is a problem for the classes $\Sigma_i(\prec)$ and $\Sigma_i(\prec, +1)$ within the quantifier alternation
hierarchies of first-order logic, as the associated classes of languages are not closed
under complement (the negation of a $\Sigma_i(\prec)$ formula is a $\Pi_i(\prec)$ formula). At first, this
was not a visible issue since the dot-depth hierarchy of star-free languages did not
include the classes $\Sigma_i(\prec, +1)$ in its original definition (see Remark 3.4).

For the purpose of describing languages, this omission is natural: one always prefers
to be allowed to use negation. However, one can argue that the logics $\Sigma_i$ are the most
fundamental ones in the two hierarchies. Indeed, the logics in the hierarchy ($\Sigma_i, \Pi_i,$
and $\Psi_i$) are all built directly from $\Sigma_i$. As we explained in the previous section, this
argument is also validated empirically as all further membership results concerning
both hierarchies are derived from the investigation of the classes $\Sigma_i(\prec)$.

The first question to be asked is whether the syntactic approach of Section 3 can be
generalized to encompass classes that are not closed under complement, such as $\Sigma_i(\prec)$. This question was answered positively by Pin [Pin 1995] and the generalized approach
was then used by Pin and Weil [Pin and Weil 1997] to obtain a membership algorithm
for $\Sigma_2(\prec)$. In this section, we explain these results.

Let $C$ be an arbitrary class of languages that satisfies the following conditions:

1. $C$ contains the languages $\emptyset$ and $A^*$ and is closed under union and intersection (but
   not necessarily under complement).
2. $C$ is closed under right and left quotients: for any $w \in A^*$ and $L \in C$
   \[w^{-1}L = \{u \mid uw \in L\} \in C\] \[Lw^{-1} = \{u \mid uw \in L\} \in C\]

   Naturally, the interesting case is when $C$ is not closed under complement, since other
cases were treated in the previous section. Let us detail why the syntactic approach
fails in this case and outline Pin’s solution to this problem.

Essentially, the relevance of the syntactic approach is justified by Theorem 3.2. As
explained, when $C$ is not closed under complement, we cannot hope to prove that $L$ is
definable in $C$ if and only if all languages recognized by its syntactic morphism are
(since these languages include the complement of $L$).

Pin’s solution to this problem was to relax the “all languages are definable” condition
in the theorem. He observed that the set of all recognized languages can be replaced by
a subset that still retains a lot of structure. To define this subset, one needs to add a
new ingredient to the mix: a canonical partial order that can be defined on the syntactic
monoid of the language. The main idea behind the definition is that given a class $C$,
whether a regular language belongs to \( C \) only depends on the syntactic morphism of the language and on this order.

**New Ingredient: The Syntactic Ordered Monoid.** One can modify the definition of the syntactic congruence of a language to define a pre-congruence (i.e., a preorder that is compatible with concatenation):

\[
u \leq_L v \quad \text{if and only if} \quad \forall x, y \in A^* \quad xuv \in L \implies xvy \in L.
\]

In turn, this defines a partial order \( \leq \) on the set of equivalence classes of \( \sim_L \) (i.e., the syntactic monoid \( M_L \) of \( L \)). It is also simple to verify that \( \leq \) is compatible with the multiplication of \( M_L \): \( s \leq t \) and \( s' \leq t' \) imply \( st \leq s't' \). This means that the pair \((M_L, \leq)\) is an ordered monoid, called the **syntactic ordered monoid** of the language.

With our goal in mind, an important observation is that while a language \( L \) and its complement \( \overline{L} \) have the same syntactic monoid \((M_L, \leq)\), they do not have the same syntactic ordered monoid. Indeed, it can be observed from the definition of the preorder \( \leq_L \) that the orders on \( M_L \) and \( M_\overline{L} \) are dual:

\[
u \leq_L v \quad \text{if and only if} \quad \forall x, y \in A^* \quad xuv \notin L \implies xvy \notin L \quad \text{if and only if} \quad v \leq_{\overline{L}} u.
\]

The following result proves that adding the syntactic ordered monoid makes it possible to recover Theorem 3.2 and the syntactic approach, even when the class \( C \) is not closed under complement. What one needs to consider is not the set of all languages possible to recover Theorem 3.2 and the syntactic approach, even when the class \( C \) is not closed under complement.

**Theorem 4.1 (Pin).** Let \( C \) be a class of languages containing \( \emptyset \) and \( A^* \) and closed under union, intersection and quotients. Then, for any regular language \( L \), \( L \) belongs to \( C \) if and only if all upward-closed languages recognized by \( \alpha_L \) belong to \( C \).

With this generalized theorem, one can hope to generalize the template established by Theorem 3.3 to classes \( C \) that are not closed under complement: deciding a semantic property of a regular language \( L \) (whether \( L \) belongs to \( C \)) can be reduced to deciding a syntactic property of the syntactic morphism \( \alpha_L \). However, this syntactic property should depend on the newly introduced order on \( M_L \). This is illustrated by the Theorem of Pin and Weil [Pin and Weil 1997], which gives a decidable characterization of \( \Sigma_2(\prec) \). In the theorem, given a word \( u \in A^* \), we denote by \( \text{alph}(u) \) its **alphabet** (i.e., the set of letters it contains).

**Theorem 4.2 (Pin-Weil).** Let \( L \) be a regular language, then the two following conditions are equivalent:

1. \( L \) is definable in \( \Sigma_2(\prec) \).
2. \( \alpha_L \) satisfies the following property, for every \( s, t \in M_L \) such that there exist two words \( u \in \alpha_L^{-1}(s) \) and \( v \in \alpha_L^{-1}(t) \) with \( \text{alph}(u) \subseteq \text{alph}(v) \):

\[
s^w \leq s^w ts^w
\]

This generalization of the syntactic approach was quite fruitful. First, by combining Theorem 4.2 with the transfer theorem of [Pin and Weil 2002], one obtains simple membership algorithms for \( \Sigma_2(\prec), \Pi_2(\prec), \Sigma_2(\prec, +1) \) and \( \Pi_2(\prec, +1) \).

**Remark 4.3.** Note that for these logics, alternate algorithms working outside of the syntactic approach are also known. An algorithm for \( \Sigma_2(\prec) \) and \( \Pi_2(\prec) \) is due to Arfi [Arfi 1987] and an algorithm for \( \Pi_2(\prec, +1) \) and \( \Pi_2(\prec, +1) \) is due to Glaßer and
Schmitz [Glaßer and Schmitz 2000]. However, these algorithms are ad hoc (they use techniques that are specific to the logic they consider), and are much harder to present.

Another success of this generalization is that it allowed to obtain a new proof of Simon’s Theorem (i.e., the decidable characterization of $\mathcal{B} \Sigma_1(<)$). While not a new result, this new proof [Henckell and Pin 2000] is of particular interest as it suggests a link between the investigation of $\Sigma_i(<)$ and that of $\mathcal{B} \Sigma_i(<)$. However, in order to formalize this relationship, one needs to consider a more general problem than membership. This is the purpose of the next section.

5. THIRD MILESTONE: SEPARATION

After the membership problem was solved for $\Sigma_2(<)$ in [Pin and Weil 1997], the next interesting classes, $\mathcal{B} \Sigma_2(<)$ and $\Sigma_3(<)$, turned out to be harder to tackle. As explained, while a partial solution for $\mathcal{B} \Sigma_2(<)$ (limited to languages over alphabets containing at most two letters) was found in [Straubing 1988], the general case remained open until very recently. In [Place and Zeitoun 2014a], membership algorithms were found for both $\mathcal{B} \Sigma_2(<)$ and $\Sigma_3(<)$.

These two results required to add a new ingredient: a new decision problem, more general than membership and called the separation problem. It is the investigation of this new problem for the classes $\Sigma_2(<)$ and $\Pi_2(<)$ that made it possible to obtain the membership algorithms for $\mathcal{B} \Sigma_2(<)$ and $\Sigma_3(<)$. Let us start with the definition of the separation problem.

5.1. The Separation Problem

The separation problem is a generalization of the membership problem. This time, we are given two input regular languages rather than one. Given three languages $L_1, L_2, K \subseteq A^*$, we say that $K$ separates $L_1$ from $L_2$ when $K$ contains $L_1$ and is disjoint from $L_2$, as shown in Figure 2.

![Fig. 2. K separates $L_1$ from $L_2$](image)

Given an arbitrary class of languages $\mathcal{C}$, the $\mathcal{C}$-separation problem is defined as follows.

Input. Two regular languages $L_1, L_2$.

Question. Is $L_1$ $\mathcal{C}$-separable from $L_2$, i.e., does there exist $K \in \mathcal{C}$ separating $L_1$ from $L_2$?

The separation problem is an immediate generalization of the membership problem: one can effectively reduce the $\mathcal{C}$-membership problem to the $\mathcal{C}$-separation problem, since a regular language belongs to $\mathcal{C}$ if and only if it is $\mathcal{C}$-separable from its complement (which is regular as well).

Note that separation is also a problem that is intuitively harder than membership. In the membership problem, one can directly perform tests on the input language to discover whether or not it belongs to $\mathcal{C}$. In contrast, for the separation problem, we are searching for a hypothetical language in $\mathcal{C}$ that is unknown at first. In fact, given a class...
C, obtaining an algorithm for the C-separation problem usually requires to consider a broader framework, more general than what we have presented so far. We do not tackle this question and refer the reader to [Place and Zeitoun 2014a; Place and Zeitoun 2014b; Place 2015] for details, and to [Almeida 1999; Henckell 1988; Henckell et al. 2010] for a view of this problem in finite semigroup theory. Instead, we concentrate on why considering this problem allowed to make progress on the membership problem for $\Sigma_3(\prec)$ and $B\Sigma_2(\prec)$. For reference, the situation for the order hierarchy updated with separation is presented in Figure 3.

![Figure 3](image-url)

**Fig. 3.** Order Quantifier Alternation Hierarchy of First-Order Logic (references are for the separation results)

The important point is that the core result in [Place and Zeitoun 2014a] is the separation algorithm for $\Sigma_2(\prec)$ and $\Pi_2(\prec)$. It is from this algorithm that membership algorithms for $\Sigma_3(\prec)$ and $B\Sigma_2(\prec)$ are derived. We illustrate this by detailing the example of $\Sigma_3(\prec)$, which is easier to present, and actually allows to transfer decidability in a generic way.

**Remark 5.1.** Our choice of presenting $\Sigma_3(\prec)$ over $B\Sigma_2(\prec)$ is mainly practical. As we will see, the membership algorithm for $\Sigma_3(\prec)$ stems from a true transfer result: membership for $\Sigma_1(\prec)$ can be *effectively* reduced to separation for $\Sigma_{i-1}(\prec)$. This makes it possible to present this result without having to present the techniques used for solving $\Sigma_2(\prec)$-separation.

On the other hand, while there is also generic connection between separation for $\Sigma_i(\prec)$ and membership for $B\Sigma_i(\prec)$ (actually even separation for $B\Sigma_i(\prec)$), it is not immediate that this connection is effective. In fact, this is open for $i \geq 3$. In particular, this means that, effectively connecting membership for $B\Sigma_2(\prec)$ to separation for $\Sigma_2(\prec)$ requires specific arguments, which turn out to be strongly tied to the $\Sigma_2(\prec)$-separation algorithm.
5.2. The Separation Problem and $\Sigma_i(<)$

We begin by explaining why considering the separation problem for $\Sigma_i(<)$ (or equivalently $\Pi_i(<)$) is relevant when considering the membership problem for $\Sigma_{i+1}(<)$. Recall the principle of the syntactic approach as outlined by Theorem 4.1: we want to use the fact that $L$ is definable in $\Sigma_3(<)$ if and only if all upward-closed languages recognized by $\alpha_L$ are. By definition, the structure of a $\Sigma_i(<)$ sentence is “layered”: the first layer is composed of $\Sigma_i(<)$ formulas, the second of $\Pi_{i-1}(<)$ formulas, the third of $\Sigma_{i-2}(<)$ formulas, and so on. The consequence of this is that deciding whether all upward-closed languages recognized by $\alpha_L$ are definable in $\Sigma_3(<)$ requires to first investigate some “$\Pi_{i-1}(<)$ problem” for these languages. Naturally, this problem has to be different and more general than membership: when $L$ is $\Sigma_i(<)$ but not $\Pi_{i-1}(<)$, we cannot hope to prove that all languages definable by $\alpha_L$ are $\Pi_{i-1}(<)$ definable.

The results of [Place and Zeitoun 2014a] state that the separation problem for $\Pi_{i-1}(<)$ is a suitable more general problem. The $\Pi_{i-1}(<)$ layer is handled by deciding which pairs of languages among those recognized by $\alpha_L$ are $\Pi_{i-1}(<)$-separable. Intuitively, this amounts to computing the “best possible $\Pi_{i-1}(<)$-definable approximation” of the languages recognized by $\alpha_L$. This is illustrated by the statement of the theorem of [Place and Zeitoun 2014a] for $\Sigma_i(<)$ membership.

**Theorem 5.2 (Place-Zeitoun).** Let $L$ be a regular language and $i \geq 2$. Then the two following conditions are equivalent:

1. $L$ is definable in $\Sigma_i(<)$.
2. $\alpha_L$ satisfies the following property, for every $s, t \in M_L$ such that $\alpha_L^{-1}(s)$ is not $\Pi_{i-1}(<)$-separable from $\alpha_L^{-1}(t)$:

   \[ s^\omega \leq s^\omega t s^\omega \]  

Theorem 5.2 states that for any $i$, the $\Sigma_i(<)$-membership problem can be effectively reduced to the $\Pi_{i-1}(<)$-separation problem. In [Place and Zeitoun 2014a], this theorem is combined with a separation algorithm for $\Sigma_2(<)$ and $\Pi_2(<)$ which yields a membership algorithm for $\Sigma_3(<)$ and $\Pi_3(<)$ (and in turn for $\Sigma_3(<,+1)$ and $\Pi_3(<,+1)$ by the transfer result of [Pin and Weil 2002]). Additionally, a separation algorithm was found recently for $\Sigma_3(<)$ and $\Pi_3(<)$ [Place 2015]. In the same way this algorithm yields membership algorithms for $\Sigma_4(<)$, $\Pi_4(<)$, $\Sigma_4(<,+1)$ and $\Pi_4(<,+1)$.

**Remark 5.3.** Note that Theorem 5.2 also provides a new $\Sigma_2(<)$-membership algorithm. However, the connection with separation was not noticed until $\Sigma_3(<)$ was investigated. The main reason for this is that the $\Pi_1(<)$-separation problem is quite simple. This simplicity entails that the condition: “$\alpha_L^{-1}(s)$ is not $\Pi_1(<)$-separable from $\alpha_L^{-1}(t)$” can be replaced by a much more elementary condition, as seen in Theorem 4.2.

6. THE FUTURE

The results of [Place and Zeitoun 2014a] seem to indicate that the membership problem might not be a general enough framework to investigate the quantifier hierarchies of first-order logic and that separation may be more appropriate. However, the situation is more complicated. After Theorem 5.2 was proved, a natural question was to know if it could be lifted to separation: can we define a problem $\mathcal{P}$, such that separation for $\Sigma_{i+1}(<)$ can be effectively reduced to $\mathcal{P}$ for $\Sigma_i(<)$? Or even better, can we choose separation as $\mathcal{P}$?

This question was investigated in [Place 2015]. While no definitive answer was found, the results of the paper suggest that the answer should be “no”. Indeed, while a separation algorithm for $\Sigma_3(<)$ and $\Pi_3(<)$ is presented in [Place 2015], the technique used to obtain it does not work by reduction to an independent problem for $\Sigma_2(<)$.
Instead, a new problem inside which the separation problems for both $\Sigma_2(<)$ and $\Sigma_3(<)$ are tied together is considered.

In short, this most likely means that the story of the quantifier alternation hierarchy is far from being over and could very well continue for several more decades.

REFERENCES


Cook proved in 1971 that SAT is NP complete, thus beginning the amazingly fruitful study of NP completeness. I’ve been teaching all my career that it is well-believed that SAT is intractable in the worst case. Over the years, however, SAT solvers have improved incredibly; they are now extensively used as general-purpose problem solvers.

In the Summer of Logic last year in Vienna, I heard an enlightening talk by Jakob Nordström on understanding the complexity of SAT using insights from Proof Complexity. I felt that this would make a great complexity column for SIGLOG News.

The following column is a valuable and readable introduction to proof complexity from the point of view of helping us understand the practical and theoretical complexity of SAT.
On the Interplay Between Proof Complexity and SAT Solving

Jakob Nordström
School of Computer Science and Communication
KTH Royal Institute of Technology
Stockholm, Sweden

This paper is intended as an informal and accessible survey of proof complexity for non-experts, focusing on some comparatively weak proof systems of particular interest in connection with SAT solving. We review resolution, polynomial calculus, and cutting planes (related to conflict-driven clause learning, Gröbner basis computations, and pseudo-Boolean solving, respectively) and some complexity measures that have been studied for these proof systems. We also discuss briefly to what extent proof complexity could provide insights into SAT solver performance, and how concerns related to applied SAT solving can give rise to interesting complexity-theoretic questions. Along the way, we highlight a number of current research challenges.

1. INTRODUCTION

The satisfiability problem (SAT) — i.e., to determine whether or not a given formula in propositional logic has a satisfying assignment — is a fundamental problem in theoretical computer science. SAT was proven NP-complete in [Cook 1971], and because of this the problem is widely believed to be computationally intractable in the worst case. Proving this currently looks totally out of reach — this is one of the million dollar Millennium Problems posed by the Clay Mathematics Institute — but it is probably fair to argue that the conventional wisdom in the computational complexity community is that SAT should be infeasible to solve in practice.

This message does not seem to have propagated to researchers in applied SAT solving, however. Instead, this area has seen enormous improvements in performance over the last two decades, and current state-of-the-art algorithms for deciding satisfiability — so-called SAT solvers — can deal with real-world instances containing millions of variables, and often run in (close to) linear time! NP-completeness did not just go away, however — it is also possible to construct tiny formulas with just a few hundred variables that are totally beyond reach for even the best solvers today.

This raises the question of how these SAT solvers work, and how they can perform so well in practice. And when they sometimes miserably fail, can one explain why?

The best current SAT solvers are based on so-called conflict-driven clause learning (CDCL). Some solvers also incorporate elements of algebraic reasoning (e.g., Gaussian elimination) and/or geometric reasoning (e.g., linear inequalities), or use algebraic or geometric methods as the foundation rather than CDCL. Another augmentation of CDCL that has attracted much interest is extended resolution. How can we analyze the power of such algorithms? The best approach we currently have is to study the underlying methods of reasoning and what they are able or unable to do in principle (mostly ignoring constructive, algorithmic aspects). This brings us into the realm of proof complexity, an area of research initiated in [Cook and Reckhow 1979].

This paper is based on the invited presentation A (Biased) Proof Complexity Survey for SAT Practitioners given at the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT ’14) [Nordström 2014].
This survey is intended as somewhat of a crash course in proof complexity, focusing on the proof systems behind some current approaches to SAT solving. We will discuss resolution (corresponding to CDCL), polynomial calculus (corresponding to algebraic Grobner basis computations) and cutting planes (corresponding to geometric or so-called pseudo-Boolean solving), and will also briefly touch on extended resolution.

Our goal is to give an overview of some of the complexity results known about these proof systems. Rather than giving precise, formal statements of theorems, our focus is on showing some of the “benchmark formulas” used to prove these theorems, since they illustrate what kind of reasoning different proof systems can or cannot do in principle. This sometimes shows fundamental limitations on what one can hope for SAT solvers to achieve, but sometimes instead can be viewed as challenges, when actual SAT solver performance does not match what theory suggests should be possible.

By necessity, this brief survey has a selective and somewhat subjective coverage of topics, and many exciting results in proof complexity are not even mentioned below. For many of the results that are mentioned, a more detailed, formal treatment can be found in the survey paper [Nordström 2013]. A more general-purpose proof complexity survey is [Segerlind 2007]. Due to space constraints, our discussion of applied SAT solving is even more limited (and sometimes slightly simplified to try to get the main message across), and we have had to omit many relevant references. A very comprehensive overview of this latter research area is given in [Biere et al. 2009].

1.1. Outline of this Survey
We review resolution in Section 2 and describe the connection to CDCL SAT solvers in Section 3. In Section 4 we discuss polynomial calculus and algebraic SAT solving, and Section 5 deals with cutting planes and geometric solvers. We comment briefly on extended resolution in Section 6. Some concluding remarks are presented in Section 7.

2. RESOLUTION
Let us start by fixing some notation and terminology:

— A literal \( a \) is a variable \( x \) or its negation \( \overline{x} \).
— A clause \( C = a_1 \lor \cdots \lor a_k \) is a disjunction of literals.
— A conjunctive normal form (CNF) formula \( F = C_1 \land \cdots \land C_m \) is a conjunction of clauses.
— A \( k \)-CNF formula is a CNF formula where all clauses contain at most \( k \) literals.

Throughout this paper, \( k \) will be some fixed constant unless stated otherwise.
— We will write \( N \) to denote the size of a formula, which is the total number of literals in it counted with repetitions (or, for a \( k \)-CNF formula, the number of clauses up to a constant linear factor).

The general set-up is that we are given an unsatisfiable CNF formula \( F \) and want to understand how efficiently the proof system under study can certify that \( F \) is contradictory. (Satisfiable formulas always have very short certificates, namely satisfying assignments, which is why it makes sense to focus on unsatisfiable instances if we want to prove complexity results.) Such a proof of unsatisfiability is often referred to as a refutation of \( F \), and we will use the two terms “proof” and “refutation” interchangeably.

We consider clauses and formulas as sets, so that there are no repetitions and order is irrelevant. In what follows, we will often tacitly assume for simplicity of exposition that the formulas involved are \( k \)-CNF formulas unless stated otherwise. There is a standard way to turn any CNF formula \( F \) into 3-CNF by converting every wide clause

\[
\begin{align*}
&\quad a_1 \lor a_2 \lor \cdots \lor a_w \\
	ag{1a}
\end{align*}
\]
into the set of 3-clauses
\[
\{ y_0 \} \cup \{ \overline{y}_{j-1} \lor a_j \lor y_j \mid 1 \leq j \leq w \} \cup \{ \overline{y}_w \},
\]
where the \( y_i \)'s denote new variables that do not appear anywhere else. This conversion to 3-CNF most often does not change much from a theoretical point of view (though there are some notable exceptions to this rule, which we will return to later).

In the resolution proof system [Blake 1937], we start with clauses of the input formula (referred to as axioms) and derive new clauses by repeated application of the resolution rule
\[
\frac{C \lor x \quad D \lor \overline{x}}{C \lor D}.
\]  
A resolution refutation ends when empty clause \( \bot \) (i.e., the clause containing no literals) has been derived. We can represent a refutation either as an annotated list of clauses as in Figure 1a or as a directed acyclic graph (DAG) as in Figure 1b. We say that a refutation is tree-like if this DAG is a tree (which is the case in Figure 1b).

It is straightforward to show that resolution is sound, i.e., that there is a resolution refutation of a formula \( F \) only if it is unsatisfiable (simply because the resolution rule (2) is sound), and it is also not hard to show that resolution is complete in that any unsatisfiable formula can be refuted.

### 2.1. Resolution Length

The length (also referred to as the size) of a resolution refutation is the number of clauses in it, counted with repetitions (which is relevant if we have a certain DAG representation of the refutation in mind, such as a tree-like refutation). In general, proof length/size is the most fundamental measure in proof complexity, and for resolution
length lower bounds also imply lower bounds on CDCL solver running time, since one can in principle extract a resolution proof from a CDCL execution trace.\footnote{This claim ignores the issue of preprocessing, which we will touch on briefly in Section 3, but a detailed discussion of which is beyond the scope of this survey.}

Any CNF formula of size $N$ can be refuted in resolution length $\exp(O(N))$, and there are formulas for which matching $\exp(\Omega(N))$ lower bounds are known. Let us discuss some examples of formulas known to be hard with respect to resolution length.

Our first example is the pigeonhole principle (PHP), which says that “$m$ pigeons do not fit into $n$ holes if $m > n$.” This is arguably the single most studied combinatorial principle in all of proof complexity (see [Razborov 2002] for a survey). When written as an unsatisfiable CNF formula, this becomes the claim that, on the contrary, $m > n$ pigeons do fit into $n$ holes. To encode this, one uses variables $p_{i,j}$ to denote “pigeon $i$ is in hole $j$,” and write down the following clauses, where $i \neq i'$ range over $1, \ldots, m$ and $j \neq j'$ range over $1, \ldots, n$:

\begin{align*}
& p_{i,1} \lor p_{i,2} \lor \cdots \lor p_{i,n} \quad \text{[every pigeon $i$ gets a hole]} \quad (3a) \\
& \overline{p}_{i,j} \lor \overline{p}_{i',j} \quad \text{[no hole $j$ gets two pigeons $i \neq i'$]} \quad (3b)
\end{align*}

There are also variants where one in addition has “functionality” and/or “onto” axioms

\begin{align*}
& \overline{p}_{i,j} \lor \overline{p}_{i,j'} \quad \text{[no pigeon $i$ gets two holes $j \neq j'$]} \quad (3c) \\
& p_{1,j} \lor p_{2,j} \lor \cdots \lor p_{m,j} \quad \text{[every hole $j$ gets a pigeon]} \quad (3d)
\end{align*}

In a breakthrough result, [Haken 1985] proved that the PHP formula consisting of clauses (3a) and (3b) requires length $\exp(\Omega(n))$ in resolution for $m = n + 1$ pigeons, and his proof can be extended to work also for the onto FPHP formulas consisting of all clauses (3a)–(3d). Later work [Raz 2004; Razborov 2003; 2004] has shown that all of the PHP formula variants remain hard even for arbitrarily many pigeons $m$, requiring resolution length $\exp(\Omega(n^\delta))$ for some $\delta > 0$. What this means, intuitively, is that the resolution proof system really cannot count — even faced with the preposterous claim that infinitely many pigeons can be mapped in a one-to-one fashion into a some finite number $n$ of holes, resolution cannot find a short proof to refute this claim.

Since PHP formulas have size $N = \Theta(n^3)$, Haken’s lower bound is only $\exp(\Omega(\sqrt{N}))$ expressed in terms of formula size. The first truly exponential lower bound on length was obtained for Tseitin formulas (an example of which is shown in Figure 2), which encode (the negation of) the principle that “the sum of the vertex degrees in a graph is even.” Here the variables are the edges in an undirected graph of bounded degree, where every vertex has been labelled 0 or 1 so that the sum of all labels is odd. Then for every vertex one writes down the set of clauses encoding that the parity of the number
of true edges incident to that vertex is equal to the vertex label (see Figure 2b, which
displays the formula corresponding to the labelled graph in Figure 2a).

If we sum the constraints over all vertices we should get an odd number by the
construction of the labelling, but since such a sum counts each edge exactly twice it has
to be even. Thus, these formulas are indeed unsatisfiable. [Urquhart 1987] established
that Tseitin formulas require resolution length \[\exp(\Omega(N))\] if the underlying graph is a
well-connected so-called expander graph (which holds asymptotically almost surely for
a random regular graph of bounded degree, for instance). Intuitively, this shows that
resolution also is not able to count \(\text{mod} \ 2\) efficiently.

Another example of exponentially hard formulas are random \(k\)-CNF formulas,
which are generated by randomly sampling \(\Delta n\) \(k\)-clauses over \(n\) variables for some
large enough constant \(\Delta\) depending on \(k\). For instance, \(\Delta \gtrsim 4.5\) is sufficient to get
unsatisfiable 3-CNF formulas asymptotically almost surely. [Chvátal and Szemerédi
1988] established that resolution requires length \(\exp(\Omega(N))\) to refute such formulas.

By now strong lower bounds have been shown for formulas encoding tiling problems
[Alekhnovich 2004], \(k\)-colourability [Beame et al. 2005], independent sets and vertex
covers [Beame et al. 2007], and many other combinatorial principles. We want to con-
clude our discussion of resolution length by mentioning perhaps the latest addition
to this long list, namely the subset cardinality formulas studied in [Spence 2010;
Van Gelder and Spence 2010; Mikša and Nordström 2014] (also known as zero-one
design or sgen formulas).

To construct these formulas, we start with an \(n \times n\) \((0,1)\)-matrix with 4 non-zero
entries in each row and column except that one extra non-zero entry is added to some
empty cell (as in Figure 3a, where the extra 1 in the bottom row is in bold face). The
variables of the formula are the non-zero entries of the matrix, yielding a total of \(4n+1\)
variables. For each row of 4 ones in the matrix, we write down the natural 3-CNF
formula encoding the positive cardinality constraint that at least 2 variables must be
true (as in the first set of clauses in Figure 3b), and for the row with 5 ones the 3-CNF
formula encoding that a strict majority of 3 variables must be true. For the columns
we instead encode negative cardinality constraints that the number of false variables is
at least 2 and 3, respectively (see the last set of clauses in Figure 3b). The formula
consisting of the conjunction of all these clauses must be unsatisfiable, since a strict
majority of the variables cannot be true and false simultaneously. We will have reason
to return to these formulas below when we discuss connections between CDCL and
resolution, and also when discussing cutting planes and pseudo-Boolean solving.

[Spence 2010; Van Gelder and Spence 2010] showed empirically that these formulas
are very hard for CDCL solvers, and [Mikša and Nordström 2014] proved the matching
theoretical result that subset cardinality formulas are indeed exponentially hard if the
underlying matrix is an expander (informally, if every at most medium-sized set of
rows has ones in many distinct columns).

2.2. Resolution Width

A second complexity measure for resolution, that is almost as well studied as length,
is width, measured as the size of a largest clause in a resolution refutation. It is clear
that the width needed to refute a formula is never larger than the number of vari-
ables \(n\), which is in turn less than the total formula size \(N\). It is also easy to see that
an upper bound \(w\) on resolution width implies an upper bound \(O(n^w)\) on resolution
length, simply because the total number of distinct clauses of width at most \(w\) over \(n\) variables is less than \((3n)^w\). Incidentally, this simple counting argument turns out to
be essentially tight, in that there are \(k\)-CNF formulas refutable in width \(w\) that require
resolution length \(n^{\Omega(w)}\), as shown by [Atserias et al. 2014].
One can also add axioms ranging over variables has a minimal element. To encode the negation of this statement in CNF, we use benchmark formula, which we describe next. 2001] showed that this is not the case by studying another interesting combinatorial strengthening (4) to something closer to (5) also for general resolution. [Bonet and Galesi not imply anything about length. This raises the question of whether it is possible to generalize resolution, however, a width lower bound even as large as of their bound in (4) for general resolution. This means that for tree-like resolution, even width lower bounds \( \omega(\log N) \) yield superpolynomial length lower bounds. For general resolution, however, a width lower bound even as large as \( \Omega(\sqrt{N \log N}) \) does not imply anything about length. This raises the question of whether it is possible to strengthen (4) to something closer to (5) also for general resolution. [Bonet and Galesi 2001] showed that this is not the case by studying another interesting combinatorial benchmark formula, which we describe next.

The ordering principle says that “every finite (partially) ordered set \( \{e_1, \ldots, e_n\} \) has a minimal element.” To encode the negation of this statement in CNF, we use variables \( x_{i,j} \) to denote “\( e_i < e_j \)” and write down the following clauses (for \( i \neq j \neq k \neq i \) ranging over \( 1, \ldots, n \)):

\[
\begin{align*}
&\tau_{i,j} \lor \tau_{j,i} & \text{[anti-symmetry; not both } e_i < e_j \text{ and } e_j < e_i \text{]} \tag{6a} \\
&\tau_{i,j} \lor \tau_{j,k} \lor x_{i,k} & \text{[transitivity; } e_i < e_j \text{ and } e_j < e_k \text{ implies } e_i < e_k \text{]} \tag{6b} \\
&\bigvee_{1 \leq i \leq n, i \neq j} x_{i,j} & \text{[} e_j \text{ is not a minimal element]} \tag{6c} \\
\end{align*}
\]

One can also add axioms

\[
\begin{align*}
&x_{i,j} \lor x_{j,i} & \text{[totality; either } e_i < e_j \text{ or } e_j < e_i \text{]} \tag{6d} \\
\end{align*}
\]

This means that if one can prove that a formula requires width \( \omega(\sqrt{N \log N}) \), this immediately yields a superpolynomial length lower bound, and a width lower bound \( \Omega(N) \) implies a truly exponential \( \exp(\Omega(N)) \) length lower bound. Almost all known lower bounds on resolution length can be derived via width lower bounds in this way (in particular, all the bounds discussed in Section 2.1, although the ones predating [Ben-Sasson and Wigderson 2001] were originally not obtained in this way).

For tree-like resolution, [Ben-Sasson and Wigderson 2001] proved a sharper version

\[
\text{tree-like refutation length } \geq 2^{\text{refutation width}} \tag{5}
\]

of their bound in (4) for general resolution. This means that for tree-like resolution, even width lower bounds \( \omega(\log N) \) yield superpolynomial length lower bounds. For general resolution, however, a width lower bound even as large as \( \Omega(\sqrt{N \log N}) \) does not imply anything about length. This raises the question of whether it is possible to strengthen (4) to something closer to (5) also for general resolution. [Bonet and Galesi 2001] showed that this is not the case by studying another interesting combinatorial benchmark formula, which we describe next.

The ordering principle says that “every finite (partially) ordered set \( \{e_1, \ldots, e_n\} \) has a minimal element.” To encode the negation of this statement in CNF, we use variables \( x_{i,j} \) to denote “\( e_i < e_j \)” and write down the following clauses (for \( i \neq j \neq k \neq i \) ranging over \( 1, \ldots, n \)):

\[
\begin{align*}
&\tau_{i,j} \lor \tau_{j,i} & \text{[anti-symmetry; not both } e_i < e_j \text{ and } e_j < e_i \text{]} \tag{6a} \\
&\tau_{i,j} \lor \tau_{j,k} \lor x_{i,k} & \text{[transitivity; } e_i < e_j \text{ and } e_j < e_k \text{ implies } e_i < e_k \text{]} \tag{6b} \\
&\bigvee_{1 \leq i \leq n, i \neq j} x_{i,j} & \text{[} e_j \text{ is not a minimal element]} \tag{6c} \\
\end{align*}
\]

One can also add axioms

\[
\begin{align*}
&x_{i,j} \lor x_{j,i} & \text{[totality; either } e_i < e_j \text{ or } e_j < e_i \text{]} \tag{6d} \\
\end{align*}
\]
to specify that the ordering is total. This yields a formula over $\Theta(n^2)$ variables of total size $N = \Theta(n^3)$. (We remark that variants of this formula also appear under the name least number principle formula or graph tautology in the literature.)

[Krishnamurthy 1985] conjectured that these formulas should be hard for resolution, but [Stålmarck 1996] showed that they are refutable in length $O(N)$ (even without the clauses (6d)). As the formula is described above, it does not really make sense to ask about the refutation width, since already the axiom clauses (6c) have unbounded width. However, one can convert the formula to 3-CNF by applying the transformation from (1a) to (1b) to the wide axioms (6c), and for this version of the formula [Bonet and Galesi 2001] established a width lower bound $\Omega(\sqrt{N})$ (which is tight, and holds even if the axioms (6d) are also added). This shows that even polynomially large resolution width does not necessarily imply any length lower bounds for general resolution.

2.3. Resolution Space

The study of the space complexity of proofs, which was initiated in the late 1990s, was originally motivated by considerations of SAT solver memory usage, but has also turned out to be of intrinsic interest for proof complexity. Space can be measured in different ways — here we focus on the most well studied measure of clause space, which is the maximum number of clauses needed in memory while verifying the correctness of a refutation. Thus, in what follows below “space” will always mean “clause space.”

The space usage of a resolution refutation at step $t$ is the number of clauses at steps $\leq t$ that are used at steps $\geq t$. Returning to our example refutation in Figure 1, the space usage at step 7 is 5 (the clauses in memory at this point are clauses 1, 3, 5, 6, and 7). The space of a proof is obtained by measuring the space usage at each step and taking the maximum. Phrased differently, one can view the formula as being stored on a read-only input tape, from where clauses can be read into working memory. The resolution rule can only be applied to clauses currently in working memory, and there is no way to store clauses “on disk” — once they are erased from working memory, they are gone. Then space measures how many clauses are used in working memory to perform the resolution refutation. Incidentally, it is not hard to see that the proof in Figure 1 is not optimal when it comes to minimizing space. We could do the same refutation in space 4 instead by processing the clauses in the order 2, 4, 6, 1, 7, 3, 5, 8, 9.

Perhaps somewhat surprisingly, an unsatisfiable CNF formula of size $N$ can always be refuted in resolution space at most $N + O(1)$, as shown by [Esteban and Torán 2001], though the refutation thus obtained might have exponential length. Lower bounds on space were subsequently shown for PHP formulas and Tseitin formulas [Alekhnovich et al. 2002; Esteban and Torán 2001] and for random $k$-CNFs [Ben-Sasson and Galesi 2003]. For the latter two formula families the (optimal linear) lower bounds matched exactly previously known width lower bounds, and also the proof techniques had a very similar flavour. This led to the question of whether there was some deeper connection hidden here.

In a very elegant paper, [Atserias and Dalmau 2008] confirmed this by showing that the inequality

$$\text{refutation space} \geq \text{refutation width} + O(1)$$  \hspace{1cm} (7)

Note, though, that this measure underestimates the actual memory usage, since storing a clause requires more than a constant amount of memory. For completeness, we mention that there is also a measure total space, counting the total number of literals in memory (with repetitions), which has been studied in [Alekhnovich et al. 2002; Bonacina et al. 2014; Bennett et al. 2015].

This space upper bound is obtained by simply running CDCL (or even DPLL) as described in Section 3 with some arbitrary but fixed variable ordering.
holds for resolution refutations of \( k \)-CNF formulas. The proof of (7) is beautiful but uses a somewhat non-explicit argument based on finite model theory. A more explicit proof, which works by simple syntactic manipulations to construct a small-width refutation from a small-space refutation, was presented in [Filmus et al. 2015a].

Since for all formulas studied up to [Atserias and Dalmau 2008] the width and space complexity measures turned out to actually coincide, it is natural to ask whether (7) could be strengthened to an asymptotic equality. The answer to this question is no. As shown in the sequence of works [Nordström 2009; Nordström and Hästad 2013; Ben-Sasson and Nordström 2008], there are formulas that can be refuted in width \( O(1) \) but require space \( \Omega(N / \log N) \) (i.e., formulas that are maximally easy for width but exhibit worst-case behaviour for space except for a log factor).

These formulas are pebbling contradictions encoding so-called pebble games on bounded fan-in DAGs, which for the purposes of this discussion we additionally require to have a unique sink. In the “vanilla version” of the formula (illustrated in Figure 4), there is one variable associated to each vertex and clauses encoding that

- the source vertices are all true;
- if all immediate predecessors are true, then truth propagates to the successor;
- but the sink is false.

There is an extensive literature on pebbling space and time-space trade-offs from the 1970s and 80s, pebbling contradictions have been useful before in proof complexity in various contexts, e.g., in [Raz and McKenzie 1999; Bonet et al. 2000; Ben-Sasson and Wigderson 2001]. Since pebbling contradictions can be shown to be refutable in constant width but there are graphs for which the pebble game requires large space, one could hope that the pebbling properties of such DAGs would somehow carry over to resolution refutations of pebbling formulas and help us separate space and width.

Unfortunately, this hope cannot possibly materialize — a quick visual inspection of Figure 4b reveals that this is a Horn formula (i.e., having at most one positive literal in each clause), and such formulas are maximally easy for length, width, and space simultaneously. However, we can modify these formulas by substituting for every variable \( x \) an exclusive or \( x_1 \oplus x_2 \) of two new variables, and then expand to CNF in the canonical way to get a new formula. This is perhaps easiest to explain by an example. Performing this substitution in the clause

\[
\begin{align*}
\pi \lor y
\end{align*}
\]

Jumping a bit ahead to Section 3 again, this is because Horn formulas are decided by unit propagation.
we obtain the formula
\[ \neg (x_1 \oplus x_2) \lor (y_1 \oplus y_2) , \] which when expanded out to CNF becomes
\[ (x_1 \lor x_2 \lor y_1 \lor y_2) \]
\[ \land (x_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2) \]  
\[ \land (\overline{x}_1 \lor x_2 \lor y_1 \lor y_2) \]  
\[ \land (\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor \overline{y}_2) . \] (8c)

With this XOR-substitution, it turns out that the pebbling contradiction inherits the
time-space trade-offs of the pebbling DAG in terms of which it is defined [Ben-Sasson
and Nordström 2008; 2011] (and there is nothing magical with XOR — this can be
shown to work also for substitution with other Boolean functions that have the “right
properties”). Now the strong space-width separation described above is obtained by
plugging in the pebbling DAGs studied in [Paul et al. 1977; Gilbert and Tarjan 1978].

2.4. Resolution Trade-offs

In the preceding sections, we have seen that for all the complexity measures of length,
width, and space there are formulas which are maximally hard for these measures.
Suppose, however, that we are given a formula that is guaranteed to be easy for two or
more of these measures. Can we then find a resolution refutation that optimizes these
complexity measures simultaneously? Or are there trade-offs, so that minimizing one
measure must cause a sharp increase in the other measure?

The first result along these lines was reported in [Ben-Sasson 2009], where a strong
space-width trade-off was established. Namely, there are formulas for which
— there are refutations in width O(1);
— there are also refutations in space O(1);
— but optimizing one measure causes (essentially) worst-case behaviour for the other.

This holds for the “vanilla version” of the pebbling contradictions in Figure 4b (if one
again uses the graphs studied in [Paul et al. 1977; Gilbert and Tarjan 1978]).

Regarding trade-offs between length and space, it was shown in [Ben-Sasson and
Nordström 2011; Beame et al. 2012; Beck et al. 2013] that there are formulas that
— can be refuted in short length;
— can be refuted in small space;
— but even slightly optimizing one measure causes a dramatic blow-up for the other.

This holds for substituted pebbling formulas over DAGs with strong time-space trade-
offs (as in, e.g., [Lengauer and Tarjan 1982]) and for Tseitin formulas over long, narrow
rectangular grids.\(^6\)

For length versus width, we know that short refutation length implies small refuta-
tion width by (4). The proof of this inequality works by transforming a given short
refutation into a narrow one, but the length blows up exponentially in the process.
Recently, [Thapen 2014] established that this blow-up is unavoidable by exhibiting
formulas for which there exist resolution refutations in short length, but for which any
refutation in width as guaranteed by (4) has to be exponentially long. These formulas
are slightly tricky to describe, however, and so we do not do so here. A technical issue

\(^6\)To be precise, the results in [Beck et al. 2013] require that one adds two copies of every edge in the grid
graph, which corresponds to XOR-substitution in the Tseitin formula, but it can be shown that this substi-
tution can be eliminated with some extra work.
with Thapen’s result is that for all other trade-offs discussed above there are \( k \)-CNF formulas that exhibit this behaviour, but Thapen’s formulas have clauses of logarithmic width. It would be nice to bring this down to constant width if possible.

3. CONNECTIONS BETWEEN RESOLUTION AND CONFLICT-DRIVEN CLAUSE LEARNING

To make the connection between the resolution proof system and conflict-driven clause learning (CDCL) SAT solvers, let us start by describing the Davis-Putnam-Logemann-Loveland (DPLL) method [Davis and Putnam 1960; Davis et al. 1962] that lies at the heart of these solvers. Given a CNF formula \( F \), this recursive method works as follows:

1. If \( F \) contains an empty clause, return \( UNSAT \).
2. If \( F \) is empty, return \( SAT \).
3. Else decide on some variable \( x \) and set it to \( true \), and make a recursive call with all satisfied clauses removed and all occurrences of \( \overline{x} \) in other clauses removed.
4. Set \( x \) to \( false \), remove satisfied clauses and occurrences of \( x \), and make a second recursive call.
5. Return \( UNSAT \) if both recursive calls returned \( UNSAT \); otherwise return \( SAT \).

How to pick the variable \( x \) in step (3) is a nontrivial question in general, and there are sophisticated decision heuristics for this, the most popular of which for CDCL solvers is VSIDS [Moskewicz et al. 2001] (which stands for variable state independent decaying sum). There is one simple special case, however: if \( F \) contains a unit clause, meaning a clause with only one literal (i.e., all other literals have been falsified by previous assignments), then we can skip the case analysis in (3) and (4) — the literal has to be set to true in order not to falsify the formula. This kind of forced assignment is known as unit propagation, and modern SAT solvers try to choose decision variables so that 99% of assignments (literally) will be unit propagations.

It is not hard to show that the search tree generated by a DPLL solver as described above can be viewed as a tree-like resolution refutation. This means that the running time of DPLL is lower-bounded by the minimum length in tree-like resolution. One problem with this is that the same work can be duplicated in different subtrees. If the DPLL solver makes a few unfortunate variable decisions at the start of the search, then it can spend a lot time exploring all combinations of assignments to the other variables in an exponential number of subtrees, when in fact the right thing to do would have been to jump back to the top and try some other assignments there.

The idea behind CDCL [Bayardo Jr. and Schrag 1997; Marques-Silva and Sakallah 1999; Moskewicz et al. 2001] is that the SAT solver should avoid such duplication of work by performing a conflict analysis when reaching a falsifying assignment and learn a reason for the failure in the form of a new clause \( C \), which is added to the formula \( F \). After this the solver has to unset variables until \( C \) is no longer falsified, and in general this might involve not just the latest decision but a whole sequence of previous decisions. This means that the solver will typically not just backtrack one level but backjump several levels. The clause \( C \) will then make sure that the SAT solver does not go into the same search subtree again. There are well-developed heuristics for how to learn clauses, and the most popular learning scheme is known as 1UIP (an abbreviation of first unique implication point). All of these learning schemes are essentially resolution derivations with very particular structural constraints.

Another feature of CDCL solvers is that they frequently clear the list of variable assignments and start the search all over again when their restart policy says so.

---

We also want to mention in this context that in a recent, very intriguing work [Razborov 2015] obtained doubly exponential size-width trade-offs in tree-like resolution (this is measured in the number of variables in the formulas, which have exponential size and polynomial width).
Restarts turn out to be very important in practice, although exactly why is poorly understood. Furthermore, since learned clauses accumulate very rapidly, solvers have a clause deletion policy that aggressively removes most of the learned clauses at regular intervals (up to 95% of the clauses over a complete run of the solver). A solver implementing all of the ideas above that has been very influential for later developments is MiniSat [Eén and Sörensson 2004], which was enhanced by [Audemard and Simon 2009] with the currently most successful heuristic for measuring which clauses to keep or delete (which can be viewed as a generalized clause width measure).

Finally, a very important aspect of modern solvers is that before the main algorithm even starts, extensive preprocessing of the input is performed, using a number of techniques that are known to be theoretically very bad in the worst case, but are completely indispensable in practice. Some solvers, such as the current SAT competitions\(^8\) champion Lingeling [Biere 2010], even interleave preprocessing and CDCL search, which is known as inprocessing [Järvisalo et al. 2012a]. Most, though not all, of these preprocessing techniques can also be formalized within resolution.

3.1. CDCL Versus Resolution

The resolution refutations found by CDCL solvers have a very specific structure. Every learned clause is derived by trivial resolution, where one keeps just one clause and resolves it sequentially with clauses currently in the clause database (i.e., axiom clauses and previously learned clauses). Moreover, the clauses used are exactly those triggering unit propagation, resolved over in reverse order of the propagations. What this means is that these refutations will look like long, sparse chains, which are tree-like except for non-local edges from previously learned clauses. This is very different from general resolution proofs, which can have very “bushy” DAG-like structure.

Since CDCL is only looking for structurally very restricted proofs, it is natural to ask how efficient CDCL proof search can be compared to the best possible general resolution proof. As mentioned above, the DPLL method corresponds to tree-like resolution, which can be exponentially worse than general resolution (see, e.g., [Ben-Sasson et al. 2004]). What about CDCL? Is it also asymptotically weaker than resolution, or could it be the case that CDCL implements efficient proof search in the sense that the method is never more than polynomially worse, say, than the shortest resolution proof?

This is a hard question, and we cannot quite expect a fully constructive affirmative answer since this would lead to collapses in parameterized complexity [Alekhnovich and Razborov 2008]. A line of works including [Beame et al. 2004; Hertel et al. 2008; Buss et al. 2008] culminated in the paper [Pipatsrisawat and Darwiche 2011] showing that CDCL viewed as a proof system polynomially simulates resolution, a result that holds for CDCL solvers with any reasonable learning scheme and restart policy. The non-constructive part is in the decision strategy, which needs to be chosen in a specific way to make the simulation go through. One possible way of interpreting this result might be that if the decision heuristic is good enough, then CDCL solvers at least have the potential to run fast on any formulas that possess short resolution proofs.

In independent work, [Atserias et al. 2011] obtained an alternative, constructive version of this result by showing that if a formula $F$ has a resolution refutation in bounded width, then CDCL using a decision strategy with enough randomness will decide $F$ efficiently. At first sight this might not seem so impressive — after all, exhaustive search in bounded width also runs fast — but the point is that a CDCL solver is very far from doing exhaustive width search and does not care at all about the existence or non-existence of narrow refutations.

\(^8\)See http://www.satcompetition.org.
The downside of both of these results is that they crucially need that the clause deletion policy is never to delete clauses. As should be clear from the discussion above, this is a very unrealistic assumption. It would be nice to extend the model of CDCL in [Atserias et al. 2011; Pipatsrisawat and Darwiche 2011] to capture memory usage in a more realistic way, and then study the question of whether CDCL can simulate resolution efficiently with respect to both time and space. A question that seems particularly interesting in this context is whether something like the theoretical time-space trade-offs in Section 2.4 could show up also in practice. The lower bounds in these trade-offs hold also in the CDCL model, but the question is whether CDCL could find resolution proofs achieving the matching upper bounds, or whether the DAG structure of these proofs are beyond anything that CDCL could possibly produce.

3.2. Theoretical Complexity Measures and Hardness in Practice

The next topic we want to discuss is whether practical hardness for CDCL is in any way related to the complexity measures of resolution length, width, and space. One interesting observation in this context is that it follows from the results reviewed in Section 2 — if we “normalize” length by taking a logarithm since it can be exponential in the formula size $N$ whereas the worst-case upper bounds for width and space are linear — that for any $k$-CNF formula the inequalities

$$\log(\text{refutation length}) \preceq \text{refutation width} \preceq \text{refutation space}$$

hold. Thus, length, width, and space form a hierarchy of increasingly strict hardness measures. Let us briefly discuss the measures again in this light:

— We know that length provides a lower bound on CDCL running time\(^9\) and that CDCL polynomially simulates resolution [Pipatsrisawat and Darwiche 2011]. However, the results in [Alekhnovich and Razborov 2008] suggest that short resolution proofs should be intractable to find in the worst case.

— Regarding width, searching for proofs in small width is apparently a well-known heuristic in the AI community, and [Atserias et al. 2011] proved that CDCL should run fast if such proofs exist.

— As to space, memory consumption is an important bottleneck for SAT solvers in practice, and space complexity results provide lower bounds on CDCL clause database size. One downside of this is that the bounds can be at most linear, and the solver would certainly use a linear amount of memory just to store the input. However, it is important to note that the space lower bounds hold even in a model where the solvers knows exactly which clauses it needs to keep. It could be argued that in reality probably much more memory than the bare minimum should be needed.

Are width or even space lower bounds relevant indicators of CDCL hardness? Or could it be true in practice that CDCL does essentially as well as resolution with respect to length/running time? These are not mathematically well-defined questions, since CDCL solvers are a moving target, but perhaps it may still be possible to perform experiments and draw interesting conclusions? Such an approach was proposed by [Ansótegui et al. 2008], and [Järvisalo et al. 2012b] performed what seems to have been the first systematic attempt to implement this program.

In view of the discussion above it seems too optimistic that length complexity should be a reliable indicator of CDCL hardness. [Järvisalo et al. 2012b] therefore focused on comparing width and space by running extensive experiments on formulas with constant width complexity (and linear length complexity) but varying space complexity to see whether running time correlated with space. These experiments produced lots

\(^9\)Except if some non-resolution-based preprocessing techniques happen to be very successful.
of interesting data, but it seems fair to say that the results are inconclusive. For some families of formulas the correlation between running time and space complexity looks very nice, but for other formulas the results seem quite chaotic.

In fact, one problem is that formulas with low width complexity and varying space complexity are hard to find — pretty much the only known examples are the substituted pebbling formulas discussed in Section 2.3. Thus, it is not even clear whether the experiments measured differences in width and space or some other property specific to these particular formulas. This problem seems inherent, however. One cannot just pick arbitrary benchmark formulas and compute the width and space complexity for them before running experiments, since deciding width is EXPTIME-complete [Berkholz 2012] and deciding space appears likely to be PSPACE-complete.

As a general comment, CDCL solver performance on combinatorial benchmark formulas is sometimes quite surprising. For instance, it seems to be folklore that if one wants to solve PHP formulas as quickly as possible, it is better to switch off the advanced heuristics that are otherwise very important for performance. For the partial ordering principle formulas in (6a)–(6c), it turns out that they are very sensitive to settings in the VSIDS decision heuristics, with small differences seemingly making all the difference between linear and exponential running time. And sometimes theoretically easy formulas are much harder than provably hard ones! For instance, for very regular matrices, such as the one in Figure 3a, subset cardinality formulas are easy even for tree-like resolution. However, even small such instances are superhard for CDCL solvers in practice (and harder than instances generated from random matrices). Again, this seems to depend on the variable decision heuristic — keeping everything else at default settings but swapping out VSIDS for a good fixed-variable decision order (based on the structure of the matrix) makes the solver run very fast.

Although CDCL performance on crafted combinatorial instances admittedly might not be immediately relevant for large-scale real-world instances, it nevertheless seems that explanations of the above phenomena could lead to a better understanding of CDCL, which is a question that is of great interest also in the applied SAT community. One important difference from industrial instances is that combinatorial benchmarks are easy to scale up and down to study the asymptotics of SAT solver behaviour, and we believe that a more systematic study of formulas such as those reviewed in Section 2 could potentially yield important insights.

4. POLYNOMIAL CALCULUS

Polynomial calculus was introduced in [Clegg et al. 1996], but we describe below a slightly modified version from [Alekhnovich et al. 2002]. In this proof system we interpret clauses as polynomials in the ring \( \mathbb{F}[x, y, z, \ldots] \), where \( x, y, z, \ldots \) are all formally distinct variables and \( \mathbb{F} \) is a field (which would be \( \text{GF}(2) \) in practical SAT solving applications but can be any field from the point of view of proof complexity).

In the context of polynomial calculus we identify 0 with true and 1 with false and translate a clause such as \( x \lor y \lor \neg z \) to the polynomial equation \( xy \neg z = 0 \), so that clauses evaluating to true corresponds to polynomials vanishing. To prove that a CNF formula is unsatisfiable, we want to show that the equations obtained from the clauses are inconsistent.

It is important to observe, however, that from an algebraic point of view \( x \) and \( \neg \) are independent variables, and also variables can take as values any elements in the

---

10 Though formulas with a “crafted flavour” turn up in, e.g., circuit verification and cryptographic problems.

11 This modified version is known as polynomial calculus (with) resolution or PCR in the literature.

12 In fact, in practical applications there would be no formal variables \( \overline{x}, \overline{y}, \overline{z} \ldots \), but they allow for a cleaner theoretical treatment of the proof system.
(a) Resolution step.  
\[
\begin{align*}
\frac{x \lor \neg y \lor z}{x \lor \neg y} & \quad \frac{y \lor z = 0}{x \lor y} \\
\frac{\neg y \lor \neg z}{\neg y \lor \neg z - \neg y = 0} & \quad \frac{\neg x y z - x y = 0}{x y z = 0} \\
\frac{x y z = 0}{-x y z + x y = 0} & \quad \frac{-x y z + x y = 0}{x y = 0}
\end{align*}
\]
(b) Corresponding polynomial calculus derivation.

Fig. 5: Example of simulation of resolution step by polynomial calculus.

Field \( \mathbb{F} \). Hence, we need to add constraints enforcing 0/1 assignments that in addition respect the meaning of negation. This leads to the following set of derivation rules:

\[
\begin{align*}
\text{Boolean axioms} & \quad \frac{x^2 - x = 0}{x + \neg x - 1 = 0} \\
\text{Negation} & \quad \frac{p = 0}{\alpha p + \beta q = 0} (\alpha, \beta \in \mathbb{F}) \\
\text{Linear combination} & \quad \frac{\neg x y z}{p = 0} \quad \frac{\neg x y z}{x p = 0} (x \text{ any variable}) \\
\text{Multiplication} & \quad \frac{x y \lor z}{q = 0}
\end{align*}
\]

A polynomial calculus refutation ends when \( 1 = 0 \) has been derived, showing that the polynomial equations have no common root, or equivalently that no assignment can simultaneously satisfy all the clauses. Polynomial calculus is sound and complete, not only for CNF formulas but for inconsistent systems of polynomial equations in general.

To define the complexity measures of size, degree, and space, we write out polynomials in a refutation as linear combinations of monomials, where we note that without loss of generality we can assume that all polynomials are multilinear (because of the Boolean axioms (10a)). Then the size of a refutation, which is the analogue of resolution length, is the total number of monomials in the refutation (counted with repetitions), the degree, corresponding to resolution width, is the largest degree of any monomial in it, and the (monomial) space, which is the analogue of resolution (clause) space, is the maximal number of monomials in memory at any point during the refutation (again counted with repetitions). One can also define a length measure for polynomial calculus, which is the number of derivation steps, but this can be exponentially smaller than the size, which is the more relevant measure to study here.

4.1. Polynomial Calculus and Resolution

Polynomial calculus can simulate resolution efficiently with respect to length/size, width/degree, and space simultaneously simply by mimicking refutations step by step. This means that all worst-case upper bounds for resolution immediately carry over to polynomial calculus. For an example of how this works, see the simulation of the resolution step in Figure 5a by the derivation in Figure 5b, where the equations corresponding to the simulated clauses are in bold face.

Polynomial calculus can be strictly stronger than resolution with respect to size and degree. For instance, over GF(2) it is not hard to see that Tseitin formulas can be refuted in size \( O(N \log N) \) and degree \( O(1) \) by doing Gaussian elimination. Another example is the set of onto FPHP formulas (3a)–(3d), which [Riis 1993] showed to be easy. It remains open, however, whether such separations can also be found for space.
**Open Problem 1.** Prove (or disprove) that polynomial calculus is strictly stronger than resolution with respect to space.

### 4.2. Polynomial Calculus Size and Degree

A lot of what is known about length versus width in resolution carries over to size versus degree in polynomial calculus. A degree upper bound $d$ implies a size upper bound $n^{O(d)}$ for refuting formulas over $n$ variables by [Clegg et al. 1996], which is qualitatively similar to the bound for resolution although the argument is a bit more involved. Just as for resolution, this bound is essentially tight by [Atserias et al. 2014].

In the other direction, a lower bound on size in terms of degree exactly analogous to the bound (4) for resolution holds for polynomial calculus as shown in [Impagliazzo et al. 1999]. Interestingly, this paper is a precursor to [Ben-Sasson and Wigderson 2001], and although it was far from obvious at the time, it turns out that one can run exactly the same proof for both of these bounds. As for resolution, the ordering principle formulas in (6a)–(6d) witness the optimality of this size-degree lower bound, as shown by [Galesi and Lauria 2010]. As for resolution, almost all size lower bounds are derived via degree lower bounds, but obtaining degree lower bounds seems much harder than width lower bounds and the machinery is much less well developed.

With the exception of Tseitin formulas and onto FPHP formulas, all the formulas in Section 2.1 are equally hard also with respect to polynomial calculus size, which can be shown via degree lower bounds arguments:

— Hardness of the standard CNF encoding (3a)–(3b) of PHP formulas follows from [Alekhnovich and Razborov 2003], with some earlier work on other non-CNF encodings in [Razborov 1998; Impagliazzo et al. 1999]. The proof in [Alekhnovich and Razborov 2003] works also if onto clauses (3d) are added, and recently [Miksa and Nordström 2015] showed that FPHP formulas with clauses (3a)–(3c) are also hard (whereas with both onto and functionality axioms added the formulas are easy, as noted above).

— Strong degree and size lower bounds on random $k$-CNF formulas were shown by [Ben-Sasson and Impagliazzo 1999] for polynomial calculus over fields of characteristic distinct from 2, and lower bounds in any characteristic including 2 were established by different methods in [Alekhnovich and Razborov 2003].

— For the subset cardinality formulas in Figure 3 [Miksa and Nordström 2014] also proved polynomial calculus degree and size lower bounds.

— Also, “Tseitin formulas with the wrong modulus” are hard — one can define Tseitin-like formulas encoding counting modulo primes $q$, and such formulas are hard over fields of characteristic $p \neq q$ [Buss et al. 2001; Alekhnovich and Razborov 2003].

### 4.3. Polynomial Calculus Space

Recall that for resolution we measure space as the number of clauses in memory, and since clauses turn into monomials in polynomial calculus the natural analogue here is monomial space. The first monomial space lower bounds were shown for PHP formulas in [Alekhnovich et al. 2002]. These formulas have wide axioms, however, and if one applies the 3-CNF conversion from (1a) to (1b) the lower bound proof breaks down.

Monomial space lower bounds for formulas of bounded width were proven only in 2012 (journal version in [Filmus et al. 2015b]) for an obfuscated 4-CNF version of PHP formulas. This was followed by optimal, linear lower bounds for random 4-CNF formulas [Bonacina and Galesi 2015], and then for Tseitin formulas over expander but with the added assumptions that either these graphs are sampled randomly or one adds two

---

13Here a twist is needed since these formulas have high initial degree, but we will not go into this.
copies of every edge to get a multigraph [Filmus et al. 2013].

Somewhat intriguingly, none of these papers could show any nontrivial lower bounds for any 3-CNF formulas. This barrier was finally overcome by [Bennett et al. 2015], who proved optimal lower bounds on random 3-CNFs. However, the following open problems show that we still do not understand polynomial calculus space very well.

OPEN PROBLEM 2. Prove polynomial calculus space lower bounds (optimal, linear bounds, or even any bounds) for Tseitin formulas over $d$-regular expander graphs for $d = 3$ or even $d > 3$ using no other assumptions than expansion only.

OPEN PROBLEM 3. Prove that PHP formulas require large monomial space in polynomial calculus even when converted to 3-CNF.

Another intriguing question is whether an analogue of the lower bound (7) on space in terms of width in resolution holds for $k$-CNF formulas also for polynomial calculus.

OPEN PROBLEM 4. Is it true that space $\geq$ degree $+ O(1)$ in polynomial calculus?

This last problem remains wide open, but [Filmus et al. 2013] made what can be described as some limited progress by showing that if a formula requires large resolution width (which is a necessary, but not sufficient, condition for high degree), then the XOR-substituted version (as in (8a)-(8c)) requires large space. When applied to Tseitin-like formulas over expander graphs, this result yields an optimal separation of space and degree. Namely, it follows that these formulas can be refuted in degree $O(1)$ but require space $\Omega(N)$. To obtain such separations we have to commit to a finite characteristic $p$ of the underlying field, however, and the formulas encoding counting $\mod p$ will separate space and degree only for fields of this characteristic. It would be nice to get a separation that would work in any characteristic, and the candidate formulas to obtain such a result readily present themselves.

OPEN PROBLEM 5. Prove (or disprove) that substituted pebbling formulas require monomial space lower-bounded by the pebbling space of the underlying DAG (which if true would yield a space-degree separation independent of the field characteristic).

4.4. Polynomial Calculus Trade-offs

When it comes to trade-offs in polynomial calculus we again recognize most of the picture from resolution, but there are also some differences:

— For space versus degree we know strong, essentially optimal trade-offs from [Beck et al. 2013], and the formulas exhibiting such trade-offs are the same vanilla pebbling contradictions as for resolution (for which we get exactly the same bounds).

— [Beck et al. 2013] also showed strong size-space trade-offs, and again the formulas used are pebbling contradictions over appropriate DAGs and Tseitin formulas over long, skinny grids. Here there is some loss in parameters as compared to resolution, however, which seems to be due to limitations of the proof techniques rather than to actual differences in formula behaviour.

— We do not yet know for sure whether the size blow-up in [Impagliazzo et al. 1999] when degree is decreased, is necessary, however, since the analysis in [Thapen 2014] works only for resolution (at least so far). This leads to the final open problem about polynomial calculus that we want to highlight in this section.

OPEN PROBLEM 6. Are there size-degree trade-offs in polynomial calculus in the sense that size has to blow up when degree is decreased in [Impagliazzo et al. 1999]?

---

14It is worth noting that these space lower bounds hold for any characteristic, so although Tseitin formulas have small-size refutations over GF(2), such refutations still require large space.
4.5. Algebraic SAT Solving

We conclude this section with a brief discussion of algebraic SAT solving. There seems to have been quite some excitement, at least in the theory community, about the Gröbner basis approach to SAT solving after the paper [Clegg et al. 1996] appeared. However, the improvement in performance that this method seemed to promise failed to materialize in practice. Instead, the CDCL revolution happened . . .

Today there are some Gröbner basis SAT solvers such as PolyBoRi [Brickenstein and Dreyer 2009; Brickenstein et al. 2009], but they do not seem competitive with resolution-based solvers. Some SAT solvers such as March successfully implement Gaussian elimination [Heule and van Maaren 2005], but this is only a very limited form of polynomial calculus reasoning.

Is it harder to build good algebraic SAT solvers than CDCL solvers? Or is it just that too little work has been done? (Witness that it took 40 years for resolution-based SAT solvers to become really efficient.) Or is it perhaps a little bit of both?

Whatever the answer is to these questions, it seems clear that one needs to find some kind of shortcut to use Gröbner bases for efficient SAT solving. A full Gröbner basis computation does too much work, since it not only decides satisfiability but yields the number of satisfying assignments, which is believed to be a strictly harder problem.

5. CUTTING PLANES

In the cutting planes proof system introduced in [Cook et al. 1987], clauses are interpreted as linear inequalities over the reals with integer coefficients, so that our example clause \( x \lor y \lor z \) gets translated to \( x + y + (1 - z) \geq 1 \), or \( x + y - z \geq 0 \) if we move all additive constants to the right-hand side (note that in contrast to polynomial calculus we now think of 1 as true and 0 as false, as we are more commonly used to).

The derivation rules are

\[
\begin{align*}
\text{Variable axioms} & \quad 0 \leq x \leq 1 \\
\text{Multiplication} & \quad \sum_i a_i x_i \geq A \\
& \quad \sum_i c a_i x_i \geq c A \\
\text{Addition} & \quad \sum_i a_i x_i \geq A \\
& \quad \sum_i b_i x_i \geq B \\
& \quad \sum_i (a_i + b_i) x_i \geq A + B \\
\text{Division} & \quad \sum_i c a_i x_i \geq A \\
& \quad \sum_i a_i x_i \geq \lfloor A/c \rfloor \\
\end{align*}
\]

where \( a_i, b_i, c, A, \) and \( B \) are all integers, and the goal is to show that a formula is unsatisfiable by deriving \( 0 \geq 1 \) from the linear inequalities corresponding to the clauses of the formula. Once again it is clear that such a derivation can exist only if the formula is indeed unsatisfiable, and the other direction of this implication also holds. We want to highlight that in the division rule (11d) we can divide with the common factor \( c \) on the left and then divide and round up the constant term on the right to the closest integer, since we know that we are only interested in 0/1 solutions. This division rule is where the power of cutting planes lies.

5.1. Cutting Planes Size, Length, and Space

The length of a cutting planes refutation is the total number of lines/inequalities in it, and the size also sums the sizes of all coefficients (i.e., the bit size of representing them). The natural generalization of clause space in resolution is to define cutting planes (line) space as the maximal number of linear inequalities needed in memory.
during a refutation, since every clause is translated into a linear inequality. There is no useful analogue of width/degree known for cutting planes.

Cutting planes can simulate resolution efficiently with respect to length/size and space simultaneously by mimicking the resolution steps one by one, and hence just as was the case for polynomial calculus we get the same worst-case upper bounds.

Cutting planes is strictly stronger than resolution with respect to length and size, since it can refute PHP formulas (3a)–(3b) efficiently [Cook et al. 1987]. The reason for this is that in contrast to resolution (and polynomial calculus), cutting planes can count. PHP formulas are refuted simply but summing up the number of pigeons and holes, after which the observation can immediately be made that there are too many pigeons to fit into the holes. It seems probable that cutting planes and polynomial calculus are incomparable with respect to size, i.e., that for both proof systems one can find hard formulas that are easy for the other system. PHP formulas show that this is true in one direction, but the other direction is open.

When it comes to space, cutting planes is very much stronger than both resolution and polynomial calculus — [Galesi et al. 2015] recently showed that any CNF formula can be refuted in constant line space \(5\) by cutting planes\(^{15}\). This proof works by starting with a linear inequality/hyperplane that cuts away the all-zero point of the Boolean hypercube \(\{0,1\}^n\) from the candidate list of satisfying assignments (there has to exist a clause falsified by this assignment, from which the hyperplane can be obtained), and then uses 4 auxiliary hyperplanes to remove further points \(\alpha \in \{0,1\}^n\) one by one in lexicographical order until all possible assignments have been eliminated, showing that the formulas is unsatisfiable. During the course of this refutation the size of the coefficients of the hyperplanes become exponentially large, however, which the line space measure does not charge for. If coefficient sizes are also counted, i.e., if one measures the total space of cutting planes refutations, then it is not hard to show a linear lower bound (for instance by combining [Ben-Sasson and Wigderson 2001] and [Beck et al. 2013]) and a quadratic worst-case upper bound is immediately implied by resolution. For resolution this quadratic upper bound is known to be tight by [Bonacina et al. 2014], but to the best of our knowledge no superlinear lower bounds are known on total space in cutting planes.

**Open Problem 7.** Are there superlinear total space lower bounds for cutting planes refutations with polynomial-size coefficients? Or with constant-size coefficients?

Proving such lower bounds, if they exist, seems challenging, however. It might be worth noting in this context that already cutting planes with coefficients of absolute size \(2\) (which is the minimum needed to simulate resolution) is quite powerful — this is sufficient to construct space-efficient refutations of PHP formulas [Galesi et al. 2015].

Essentially the only formulas that are known to be hard for the cutting planes proof system with respect to length/size are the **clique-coclique formulas** claiming (the negation of) that "a graph with an \(m\)-clique cannot be \((m-1)\)-colourable." The formulas consist of clauses:

\[
\begin{align*}
q_{k,1} \lor q_{k,2} \lor \cdots \lor q_{k,n} & \quad \text{[some vertex is the \(k\)th member of the clique]} \quad (12a) \\
\neg q_{k,i} \lor \neg q_{k,j} & \quad \text{[the \(k\)th clique member is uniquely defined]} \quad (12b) \\
p_{i,j} \lor \neg p_{i,j} & \quad \text{[clique members are connected by edges]} \quad (12c) \\
r_{i,1} \lor r_{i,2} \lor \cdots \lor r_{i,m-1} & \quad \text{[every vertex \(i\) has a colour]} \quad (12d) \\
\neg r_{i,j} \lor \neg r_{j,t} & \quad \text{[neighbours have distinct colours]} \quad (12e)
\end{align*}
\]

\(^{15}\)Recall that this means that the formula is kept on a read-only input tape, and the working memory never contain more than 5 inequalities at any given time.
where variables $p_{i,j}$ are indicators of the edges in an $n$-vertex graph, variables $q_{k,i}$ identify the members of an $m$-clique in the graph, and variables $r_{i,t}$ specify a colouring of the vertices, for indices ranging over $1 \leq i \neq j \leq n$, $1 \leq k \neq k' \leq m$, and $1 \leq t \leq m-1$.

[Pudlák 1997] proved that these formulas are hard by using a so-called interpolation argument, specifically tailored to work for formulas with the right structure. He showed that from any short cutting planes refutation of the formula, one can extract a small monotone circuit for clique, which reduces the problem to a question about proving size lower bounds for monotone circuits.

It seems plausible that the Tseitin formulas in Figure 2 should require long cutting planes refutations, since it should be hard to count mod 2 using linear inequalities, and if this could be shown it would follow that cutting planes and polynomial calculus are incomparable with respect to proof size. It also seems very likely that random $k$-CNF formulas should be very hard, but no such lower bounds are known, nor any other lower bounds not using the interpolation method. These are all longstanding open problems in proof complexity.

**Open Problem 8.** Prove length lower bounds for cutting planes refutations of Tseitin formulas or random $k$-CNF formulas, or for any formula family by using some technique other than interpolation.

### 5.2. Size-Space Trade-offs for Cutting Planes

Given our very limited understanding of cutting planes, it is perhaps not so surprising that not very much is known about size-space trade-offs for this proof system.

[Göös and Pitassi 2014] showed that short cutting planes refutations of Tseitin formulas on expanders must have large space, but this does not provide a real trade-off since it seems likely that such short refutations do not exist at all, regardless of their space complexity. [Huynh and Nordström 2012] proved that short cutting planes refutations of one particular version of pebbling contradictions (slightly different from the substituted pebbling contradictions discussed in Section 2.3) over one particular family of DAGs require large space — a result that was strengthened and generalized by [Göös and Pitassi 2014] — and for pebbling contradictions such short refutations do exist. Interestingly, and somewhat unexpectedly, all of these results follow from reductions to communication complexity. The state of knowledge regarding pebbling contradictions is much worse here than for resolution and polynomial calculus, however — for the latter two proof systems we know of general methods to translate pebbling trade-offs for (essentially) arbitrary graphs into proof complexity size-space trade-offs.

Since [Galesi et al. 2015] established that any CNF formula has a a constant-space refutation, the lower bounds for pebbling contradictions in [Huynh and Nordström 2012; Göös and Pitassi 2014] yield true size-space trade-off results for cutting planes, with formulas that can be refuted in both small size and small space, but where optimizing both measures simultaneously is impossible. However, the “space-efficient” refutations have coefficients of exponential size. It would be more convincing to obtain trade-offs where the small-space refutations also have small coefficients (which would follow if known resolution and polynomial calculus results for pebbling contradictions or Tseitin formulas over long, skinny grids could be lifted also to cutting planes).

**Open Problem 9.** Are there trade-offs where the space-efficient CP refutations have small coefficients (say, of polynomial or even constant size)?

### 5.3. Pseudo-Boolean SAT Solving

Work on so-called pseudo-Boolean solvers exploring (a subset of) cutting planes has been done by, e.g., [Dixon and Ginsberg 2002; Dixon et al. 2004; Chai and Kuehlmann 2005; Sheini and Sakallah 2006; Le Berre and Parra, 2010], but it seems like a very
ones with positive coefficients) over all vertices to derive is odd, however, then cutting planes can sum the at-least-Eulerian cycle and label every second edge number of edges in the graph is even, then this formula is satisfiable — just fix any every vertex and writes down constraints that edges should be labelled with a connected graph.

The family of benchmark formulas in this context are the formulas intriguinig family of benchmark formulas in this context are the cardinality constraints is probably not an efficient approach in general.

This is not the only challenge for pseudo-Boolean solvers, however. Another quite intriguing family of benchmark formulas in this context are the even colouring (EC) formulas (satisfiable instance).

tough challenge to make these solvers as efficient as CDCL solvers. This is underlined by the fact that a competitive option for pseudo-Boolean solving is to simply translate the input to CNF and run a CDCL solver, as shown by [Eén and Sörensson 2006].

As it turns out, one problem with current pseudo-Boolean solvers is that if they get their input in CNF, they cannot even go beyond resolution. Solvers such as Sat4j [Le Berre and Parrain 2010] solve PHP formulas very efficiently, but they crucially depend on the input being given as linear inequalities:

\[
\begin{align*}
p_{i,1} + p_{i,2} + \cdots + p_{i,n} &\geq 1 \quad &\text{[every pigeon } i \text{ gets a hole]} \tag{13a} \\
-p_{1,j} - p_{2,j} - \cdots - p_{n+1,j} &\geq -1 \quad &\text{[no hole } j \text{ gets two pigeons]} \tag{13b}
\end{align*}
\]

If the input is instead presented in CNF, with the cardinality constraints in Equation (13b) encoded as the clauses in Equation (3b), then Sat4j runs in exponential time. The same holds for subset cardinality formulas — if a pseudo-Boolean solver is fed the formula encoded as cardinality constraints, then it runs fast, but on the CNF version in Figure 3b it cannot possibly do better than the exponential lower bound on resolution length in [Mikša and Nordström 2014].

Thus, an algorithmic challenge is to make pseudo-Boolean solvers reason more efficiently with CNF inputs, so that they could, e.g., detect and use the cardinality constraints hidden in (3a)–(3b) to get performance comparable to when the input is given as (13a)–(13b). It is possible to do a preprocessing step to recover cardinality constraints encoded in CNF, and for PHP formulas and subset cardinality formulas this works well [Biere et al. 2014], but full preprocessing of the input to try to detect cardinality constraints is probably not an efficient approach in general.

This is not the only challenge for pseudo-Boolean solvers, however. Another quite intriguing family of benchmark formulas in this context are the even colouring (EC) formulas constructed by [Markström 2006] and shown in Figure 6. Here one starts with a connected graph \( G \) having an Eulerian cycle, i.e., with all vertex degrees even, and writes down constraints that edges should be labelled 0/1 in such a way that for every vertex \( v \) in \( G \) the number of 0-edges and 1-edges incident to \( v \) is equal. If the total number of edges in the graph is even, then this formula is satisfiable — just fix any Eulerian cycle and label every second edge 0 and 1, respectively. If the number of edges is odd, however, then cutting planes can sum the at-least-2 constraints in Figure 6c (the ones with positive coefficients) over all vertices to derive \( 2 \cdot \sum_{e \in E(G)} e \geq |E(G)| \) and then

\[
\begin{align*}
\frac{\sum_{i} p_{i,1} + p_{i,2} + \cdots + p_{i,n}}{2} &\geq 1 \\
\frac{-\sum_{j} p_{1,j} - p_{2,j} - \cdots - p_{n+1,j}}{2} &\leq -1
\end{align*}
\]

\[
\begin{align*}
(u \lor w) \land (w \lor x \lor y) \land (u \lor z) \land (z \lor x \lor y) \\
\land (v \lor y) \land (v \lor \overline{y} \lor z) \land (v \lor \overline{y} \lor w) \land (v \lor \overline{y} \lor y)
\end{align*}
\]

(a) Eulerian graph. (b) Corresponding CNF formula. (c) Pseudo-Boolean encoding.

Fig. 6: Example of Markström’s even colouring (EC) formula (satisfiable instance).
divide by 2 and round up to obtain \( \sum_{e \in E(G)} e \geq (|E(G)| + 1)/2 \). By instead summing up all at-most-2 constraints (the ones with negative coefficients) and dividing by 2 one obtains \( \sum_{e \in E(G)} e \leq (|E(G)| - 1)/2 \), and subtracting these two inequalities yields \( 0 \geq 1 \).

One interesting aspect to observe here is that in contrast to PHP and subset cardinality formulas, the above argument uses crucially that variables are integer-valued. To see the difference, suppose that we are given a PHP or subset cardinality formula encoded as linear constraints. Then for cutting planes it is sufficient to simply add up the inequalities to derive a contradiction. No integer-based reasoning is needed. Even if we allow putting fractional pigeons into fractional holes, there is no way one can make a pigeon mass of \( n + 1 \) fit into holes of total capacity \( n \). This set of linear inequalities is unsatisfiable even over the rationals, i.e., the polytopes defined by the constraints is empty. Similarly, for subset cardinality formulas there is no way \( 4n + 1 \) variables could have a total “true mass” of at least \( 2n + 1 \) and a total “false mass” of \( 2n + 1 \) simultaneously. But for collections of linear constraints as in Figure 6c, assigning all edges value \( \frac{1}{2} \) is a satisfying fractional solution. The polytope defined by the linear inequalities is not empty, but it does not contain any integer points. Hence, refuting EC formulas in cutting planes crucially requires the division rule (11d), and pseudo-Boolean solvers need to implement this rule (or some other form of integer-based reasoning) to decide these formulas efficiently. Based on some limited experiments, however, EC formulas appear to be much harder for pseudo-Boolean solvers than the cutting planes upper bound would suggest, which seem to indicate that the solvers are still quite far from using the full power of cutting planes reasoning.

6. EXTENDED RESOLUTION

A topic of considerable interest in the applied SAT community lately has been if and how SAT solvers can be enhanced to use extended resolution, an approach that was proposed in, e.g., [Sinz and Biere 2006; Audemard et al. 2010]. What this means is that one starts with a CDCL solver searching for resolution proofs as usual, but adds the option of introducing new variables as names of subformulas. What can proof complexity say about such an approach to SAT solving?

Not too much — if there are no restrictions on how these new variables can be added, then this corresponds to the proof system extended Frege, which is an extremely strong proof system for which essentially no lower bounds are known. So this means that the potential gains in performance are enormous, but of course the question is how the new variables should be added to realize this potential.

In order to make it possible to study the kind of extended resolution used in CDCL solvers, one needs to describe in more detail the rules actually used for adding new variables. Once one has such a description, such as, e.g., for the bounded variable addition used in [Manthey et al. 2013], it is possible to reason about what such a limited form of extended resolution could or could not do (although proving formal theorems about this might still be a formidable problem).

7. CONCLUDING REMARKS

In this column, we have presented an overview of the proof systems resolution, polynomial calculus and cutting planes, motivated by the fact that these systems serve as foundations for SAT solvers using conflict-driven clause learning (CDCL), algebraic Gröbner basis computations, and pseudo-Boolean techniques, respectively. The discussion has intentionally been kept at a high level, with only informal statements of results. For many of the proof complexity results mentioned in this paper it is possible to find exact, formal statements in the survey paper [Nordström 2013].
On the proof complexity side, the main take-away message is that resolution is fairly well understood, although there are still some interesting open questions left (which we mostly did not discuss). For polynomial calculus we also have a fair amount of knowledge, although there are many more open problems than for resolution. For instance, the techniques for proving degree lower bounds (and hence size lower bounds) are not yet very well developed, and the hardness status of several interesting formula families remain open. Also, we do not understand very well the relations between degree and monomial space. For cutting planes, almost nothing is known, and any progress on the open problems listed in this survey would be very exciting.

When it comes to applied SAT solving, we still have quite a poor understanding of why different formulas are easy or hard. It would be interesting to investigate further whether there could be any relevant connections here between proof complexity measures and hardness of SAT, or whether tools and techniques from proof complexity could help to shed light on the inner workings on SAT solvers.

Finally, the main algorithmic challenge we want to highlight is if and how one can build efficient SAT solvers based on stronger proof systems than resolution. Is it really the case that CDCL, originating in the DPLL method from the early 1960s [Davis and Putnam 1960; Davis et al. 1962; Robinson 1965], is the best conceivable paradigm? Or could it be possible that it is now time, over 50 years later, to take the next step and build fundamentally different SAT solvers based on algebraic and/or geometric methods? Are there perhaps fundamental limitations why efficient proof search cannot be implemented within these proof systems? Or could it be that a sustained long-term effort would yield powerful new SAT solving paradigms, just as the immense work spent on optimizing CDCL solvers over the years have led to improvements in performance of several orders of magnitude?

ACKNOWLEDGMENTS

Many colleagues in the proof complexity community have shaped my thinking on this topic, and I am probably more indebted to them than I myself realize. In addition, in connection with this survey I want to acknowledge all the SAT researchers who have kindly listened to my never-ending stream of questions over the last few years and have patiently answered them (sometimes repeatedly). I am especially thankful to Gilles Audemard, Armin Biere, Niklas Een, Marijn Heule, Matti Järvisalo, Daniel Le Berre, João Marques-Silva, Karem Sakallah, Laurent Simon, and Niklas Sörensson for many helpful discussions.

I wish to thank Neil Immerman for asking me to write this survey and gently nudging me to follow through. I am very grateful to him as well as to Christoph Berkholz, Armin Biere, Ilario Bonacina, Marijn Heule, Matti Järvisalo, and Massimo Lauria for helping to proof-read the manuscript, suggesting many improvements and discovering many inaccuracies in the process. It goes without saying that any opinions expressed, and any errors remaining, are the responsibility of the author only.

The author was funded by the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no. 279611 as well as by Swedish Research Council grants 621-2010-4797 and 621-2012-5645.

REFERENCES


Ilario Bonacina, Nicola Galesi, and Neil Thapen. 2014. Total Space in Resolution. In

Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. 2000. On the Relative Complexity


Ilario Bonacina, Nicola Galesi, and Neil Thapen. 2014. Total Space in Resolution. In

Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. 2000. On the Relative Complexity


Ilario Bonacina, Nicola Galesi, and Neil Thapen. 2014. Total Space in Resolution. In

Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. 2000. On the Relative Complexity


Ilario Bonacina, Nicola Galesi, and Neil Thapen. 2014. Total Space in Resolution. In

Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. 2000. On the Relative Complexity


Ilario Bonacina, Nicola Galesi, and Neil Thapen. 2014. Total Space in Resolution. In

Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. 2000. On the Relative Complexity


Differential privacy is one of the most popular privacy definitions in the modern literature: originally proposed to protect the privacy of individual users in statistical databases, it was recently leveraged in the context of anonymity networks, web analytics, online advertising, and many other systems. In fact, differential privacy constitutes an interdisciplinary research field, combining expertise from various communities, such as statistics, cryptography, databases, distributed systems, and program verification.

Before exploring the link between differential privacy and program verification, I thought it would have been useful to introduce in this column the concept of differential privacy and to illustrate its practical relevance through a concrete application scenario. I am glad that Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati accepted my invitation to report on their exciting work on geoidistinguishability, which leverages differential privacy to protect the privacy of users when interacting with location-based services. This ongoing project has already delivered not only deep theoretical results, but also a practical extension for desktop and mobile browsers to automatically protect location information in the Internet.

A big thanks to the authors for accepting the invitation and for their awesome contribution.
Location Privacy via Geo-indistinguishability

Konstantinos Chatzikokolakis
CNRS and
École Polytechnique, France

Catuscia Palamidessi
INRIA and
École Polytechnique, France

Marco Stronati
École Polytechnique, France

In this paper we report on our ongoing project aimed at protecting the privacy of the user when dealing with location-based services. The starting point of our approach is the principle of geo-indistinguishability, a formal notion of privacy that protects the user’s exact location, while allowing approximate information – typically needed to obtain a certain desired service – to be released. We then present two mechanisms for achieving geo-indistinguishability, one generic to sanitize locations in any setting with reasonable utility, the other custom-built for a limited set of locations but providing optimal utility. Finally we extend our mechanisms to the case of location traces, where the user releases his location repeatedly along the day and we provide a method to limit the degradation of the privacy guarantees due to the correlation between the points. All the mechanisms were tested on real datasets and compared both among themselves and with respect to the state of the art in the field.

1. INTRODUCTION
The widespread use of Location-Based Services (LBS) in today’s world has created new risks to user privacy that users are increasingly becoming aware of. In large part, the worries are caused by the shocking episodes of violations and leaks that keep appearing on the news. Just to mention a couple of them, on April 20th, 2011 it was discovered that the iPhones were storing and collecting location data from their users, syncing them with iTunes and transmitting them to Apple, all without the users’ knowledge. More recently, the Guardian has revealed, on the basis of the documents provided by Edward Snowden, that the NSA and the GCHQ have been using certain smartphone apps, such as the wildly popular Angry Birds game, to collect users’ private information such as age, gender and location [Ball 2014].

To some extent, also the research and the experimentation on privacy contribute to raise the awareness about the practical risks. For instance, the “Please Rob Me” website [http://pleaserobme.com/] aggregates location check-ins and presents them as “robbery opportunities”, pointing out the fact that publically announcing one’s location effectively reveals to the world that they are not home.

A survey among 180 smartphone users, described by Fawaz and Shin [2014], reported that 78% of the participants believe that apps accessing their location can pose privacy threats. Furthermore, 85% of them declared that they care about who accesses their location information. All these worries about location privacy may seem exaggerated at first, but one can see that they are fully justified when thinking to the possible malicious uses of location information, such as robbing and stalking. For instance, the application “Girls Around Me”, combines social media and location information to find nearby women (who hadn’t necessarily agreed to be found), and, with one click the
user can access the Facebook profiles of targeted girls [Brownlee 2012]. Particularly worrisome is the perspective of potential combination with the users’ most sensitive information, such as sexual orientation. Again, according to the Guardian [Ball 2014], there have been cases of smartphone applications from which such information was collected without the user’s knowledge.

Furthermore, location information can be easily used to obtain a variety of other information that an individual usually wishes to protect: by collecting and processing accurate location data on a regular basis, it is possible to infer an individual’s home or work location, sexual preferences, political views, religious inclinations, etc.

There are numerous programs that collect location data from mobile devices. In this paper, we focus our attention to those applications which collect such data to provide an agreed-upon service, i.e., the LBSs. Obviously there exist methods for preventing the collection of location data entirely, however they would completely nullify the benefits of applications which provide location services. Our primary goal is to develop methods that hinder the undesired tracking capacities of LBSs, while preserving as much as possible the quality of the desired services.

Several notions of privacy for location-based systems have been proposed in the literature. In Section 2 we give an overview of such notions, and we discuss their shortcomings in relation to our motivating LBS applications. Aiming at addressing these shortcomings, we propose a formal privacy definition, called geo-indistinguishability, that allows a user to disclose enough location information to obtain the desired service, while satisfying the aforementioned privacy notion. Our proposal is based on a generalization of differential privacy [Dwork 2006] developed by Chatzikokolakis et al. [2013]. Similarly to differential privacy, our notion and technique abstract from the side information of the adversary, such as any prior probabilistic knowledge about the user’s actual location.

To explain the principle of geo-indistinguishability, consider a user located in Paris who wishes to query an LBS provider for nearby restaurants in a private way. To achieve this the user employs obfuscation, i.e. he discloses some approximate location $z$ instead of his exact one $x$. Interestingly, 52% of the surveyed individuals in [Fawaz and Shin 2014] stated no problem in supplying apps with imprecise location information to protect their privacy; only 18% objected to providing imprecise location information. Note that, in contrast to various works in the literature, we assume that the user is interested in hiding his location, not his identity; in fact, the user might be authenticated to the service provider in order to obtain personalized recommendations.

We say that the user enjoys $\ell$-privacy within $r$ if, any two locations at distance at most $r$ produce observations with “similar” distributions, where the “level of similarity” depends on $\ell$. The idea is that $\ell$ represents the user’s level of privacy for that radius: the smaller $\ell$ is, the higher is the privacy.

The definition of geo-indistinguishability abstracts from $r$ by requiring that the (inverse of the) level of privacy $\ell$ depend on the radius $r$. Formally: A mechanism satisfies geo-indistinguishability iff for any radius $r > 0$, the user enjoys $\ell r$-privacy within $r$.

This definition implies that the user is protected within any radius $r$, but with a level $\ell = \epsilon r$ that increases with the distance. Within a short radius, for instance $r = 1$ km, $\ell$ is small, guaranteeing that the provider cannot infer the user’s location within, say, the 7th arrondissement of Paris. Farther away from the user, for instance for $r = 1000$ km, $\ell$ becomes large, allowing the LBS provider to infer that with high probability the user is located in Paris instead of, say, London.

We propose a mechanism that achieves geo-indistinguishability by perturbing the user’s location $x$. The inspiration for our mechanism comes from one of the most popular approaches for differential privacy, namely the Laplace noise. We adopt a specific planar version of the Laplace distribution, allowing to draw points in a geo-
indistinguishable way; moreover, we are able to do so efficiently, by using polar coordinates. Another advantage of the resulting mechanism is that it is independent from the particular user or the area it is used in, the only parameter is the desired level of privacy or conversely the desired level of accuracy of the service.

Clearly, the perturbation of the information sent to the LBS provider leads to a degradation of the quality of service, and consequently there is a trade-off between the level of privacy that the user wishes to guarantee and the service quality loss (QL) that he has to accept. The study of this trade-off, and the design of mechanisms which optimize it, is an important research direction started with the seminal paper by Shokri et al. [2011]. Andrés et al [2013] have compared our mechanism with other ones in the literature, using the privacy metric proposed in [Shokri et al. 2012]. It turns our that our mechanism offers the best privacy guarantees, for the same utility, among those which do not depend on the user.

The advantages of the independence from the user are obvious: first, the mechanism is designed once and for all, we do not need different mechanisms for different users. Second, even the same user may have different behaviors, for instance during different parts of the day, and it would not be practical to change the mechanism all the time. Finally, computing the prior of the user can be an expensive operation, and in some cases even unfeasible.

However, if we are interested in protecting a particular user, then in general there are mechanisms, specific for that user, that do better than the generic Laplace mechanism. Thus, we are also interested in defining specialized mechanisms that optimize the trade-off between geo-indistinguishability and quality of service for a particular user. More precisely, given a certain threshold on the degree of geo-indistinguishability, and a prior, we aim at obtaining the mechanism \( K \) which minimizes the QL. Based on the fact that the geo-indistinguishability threshold can be expressed by linear constraints, we can reduce the problem of producing such an optimal \( K \) to a linear optimization problem, which can then be solved by using standard techniques of linear programming.

The two mechanisms discussed above correspond to a sporadic use of the service in which a single location needs to be sanitized. In practice, however, a user might performs repeated location-based queries from several locations, forming a location trace that he wishes to protect. For each query, a new obfuscated location needs to be reported to the service provider, which can be easily obtained by independently adding noise at the moment when each query is executed. We refer to independently applying noise to each location as the independent mechanism.

However, it is easy to see that privacy is degraded as the number of queries increases, due to the correlation between the locations. Intuitively, in the extreme case when the user never moves (i.e. there is perfect correlation), the reported locations are centered around the real one, thus revealing it more and more precisely as the number of queries increases. Technically, the independent mechanism applying \( \epsilon \)-geo-indistinguishable noise (where \( \epsilon \) is a privacy parameter) to \( n \) locations can be shown to satisfy \( n\epsilon \)-geo-indistinguishability. This is typical in the area of differential privacy, in which \( \epsilon \) is thought as a privacy budget, consumed by each query; this linear increase makes the mechanism applicable only when the number of queries remains small. In order to deal with multiple queries we propose a trace obfuscation mechanism with a smaller budget consumption rate than applying independent noise [Chatzikokolakis et al. 2014]. The main idea is to actually use the correlation from previous locations to try to predict a point close to the user’s actual location. Predicted points are safe to report directly and thus have a smaller footprint on the privacy budget.

We experimentally compare the above mechanisms on two large real-life data sets, Geolife and Tdrive. The results show the utility improvements of the optimal con-
structured mechanism wrt the Laplace one, as well as the improvements of the predictive mechanism wrt the independently applied noise.

This paper presents a systematic overview of the approach to location privacy developed by our INRIA team Comète. Some of the results presented here have appeared in previous papers of ours specialized in particular aspects of the project [Andrés et al. 2013; Chatzikokolakis et al. 2014; Bordenabe et al. 2014].

Road Map. In Section 2 we discuss notions of location privacy from the literature and point out their weaknesses and strengths. In Section 3 we formalize the notion of geo-indistinguishability in three equivalent ways. We then proceed to describe two mechanisms that provide geo-indistinguishability in Section 4: one general, the other with optimal utility. In Section 5 we propose a predictive mechanism that exploits correlations on the input by means of a prediction function to improve the privacy guarantee. In Section 6 we give an overview of the experimental analysis and comparison of the mechanisms. Section 7 presents Location Guard, a tool based on Geo-indistinguishability. Section 8 presents the ongoing work. Section 9 presents the related work, and Section 10 concludes.

2. EXISTING NOTIONS OF PRIVACY

In this section, we examine various notions of location privacy from the literature, as well as techniques to achieve them. We consider the motivating example from the introduction, of a user in Paris wishing to find nearby restaurants with good reviews. To achieve this goal, he uses a handheld device (e.g., a smartphone) to query a public LBS provider. However, the user expects his location to be kept private: informally speaking, the information sent to the provider should not allow him to accurately infer the user’s location. Our goal is to provide a formal notion of privacy that adequately captures the user’s expected privacy. From the point of view of the employed mechanism, we require a technique that can be performed in real-time by a handheld device, without the need of any trusted anonymization party.

Expected Adversary Error. The expected error of an optimal Bayesian adversary [Shokri et al. 2011; Shokri et al. 2012; Hoh and Gruteser 2005] is a natural way to quantify the privacy offered by a location-obfuscation mechanism. Intuitively, it reflects the degree of accuracy by which an adversary can guess the real location of the user by observing the obfuscated location, and using any side-information available to him.

There are several works relying on this notion. In [Hoh and Gruteser 2005], a perturbation mechanism is used to confuse the attacker by crossing paths of individual users, rendering the task of tracking individual paths challenging. In [Shokri et al. 2012], an optimal location-obfuscation mechanism (i.e., achieving maximum level of privacy for the user) is obtained by solving a linear program in which the constraints are determined by the quality of service and by the user’s profile. In [Herrmann et al. 2013] bandwidth constraints are also taken into account, while Theodorakopoulos et al. [2014] consider the case of repeated location reporting, as opposed to a sporadic use of the mechanism. Furthermore, Olteanu et al. [2014] analyze the case where the attacker can also exploit co-location information, such as geo-located pictures, shared on a social network, in which several friends are tagged together.

It is worth noting that this privacy notion and the obfuscation mechanisms based on it are explicitly defined in terms of the adversary’s side information. In contrast, our notion of geo-indistinguishability abstracts from the attacker’s prior knowledge, and is therefore suitable for scenarios where the prior is unknown, or the same mechanism must be used for multiple users.
The notion of $k$-anonymity is the most widely used definition of privacy for location-based systems in the literature. Many systems in this category [Gruteser and Grunwald 2003; Gedik and Liu 2005; Mokbel et al. 2006] aim at protecting the user's identity, requiring that the attacker cannot infer which user is executing the query, among a set of $k$ different users. Such systems are outside the scope of our problem, since we are interested in protecting the user’s location.

On the other hand, $k$-anonymity has also been used to protect the user’s location (sometimes called $l$-diversity in this context), requiring that it is indistinguishable among a set of $k$ points (often required to share some semantic property). One way to achieve this is through the use of dummy locations [Kido et al. 2005; Shankar et al. 2009]. This technique involves generating $k - 1$ properly selected dummy points, and performing $k$ queries to the service provider, using the real and dummy locations. Another method for achieving $k$-anonymity is through cloaking [Bamba et al. 2008; Duckham and Kulik 2005; Xue et al. 2009]. This involves creating a cloaking region that includes $k$ points sharing some property of interest, and then querying the service provider for this cloaking region.

Even when side knowledge does not explicitly appear in the definition of $k$-anonymity, a system cannot be proven to satisfy this notion unless assumptions are made about the attacker’s side information. For example, dummy locations are only useful if they look equally likely to be the real location from the point of view of the attacker. Any side information that allows to rule out any of those points, as having low probability of being the real location, would immediately violate the definition.

Counter-measures are often employed to avoid this issue: for instance, Kido et al. [2005] take into account concepts such as ubiquity, congestion and uniformity for generating dummy points, in an effort to make them look realistic. Similarly, Xue et al. [2009] take into account the user’s side information to construct a cloaking region. Such counter-measures have their own drawbacks: first, they complicate the employed techniques, also requiring additional data to be taken into account (for instance, precise information about the environment or the location of nearby users), making their application in real-time by a handheld device challenging. Moreover, the attacker’s actual side information might simply be inconsistent with the assumptions being made.

As a result, notions that abstract from the attacker’s side information, such as differential privacy, have been growing in popularity in recent years, compared to $k$-anonymity-based approaches.

Differential Privacy. Differential Privacy [Dwork 2006] is a notion of privacy from the area of statistical databases. Its goal is to protect an individual's data while publishing aggregate information about the database. Differential privacy requires that modifying a single user's data should have a negligible effect on the query outcome. More precisely, it requires that the probability that a query returns a value $v$ when applied to a database $D$, compared to the probability to report the same value when applied to an adjacent database $D'$ – meaning that $D, D'$ differ in the value of a single individual – should be within a bound of $e^\epsilon$. A typical way to achieve this notion is to add controlled random noise to the query output, for example drawn from a Laplace distribution. An advantage of this notion is that a mechanism can be shown to be differentially private independently from any side information that the attacker might possess.

Differential privacy has also been used in the context of location privacy. In [Machanavajjhala et al. 2008], it is shown that a synthetic data generation technique can be used to publish statistical information about commuting patterns in a differentially private way. In [Ho and Ruan 2011], a quadtree spatial decomposition technique is used to ensure differential privacy in a database with location pattern mining ca-
pabilities, while Chen et al. [2012] use variable-length $n$-grams to disclose sequential data, such as mobility traces, in a differentially private way.

As shown in the aforementioned works, differential privacy can be successfully applied in cases where aggregate information about several users is published. On the other hand, the nature of this notion makes it poorly suitable for applications in which only a single individual is involved, such as our motivating scenario. The secret in this case is the location of a single user. Thus, differential privacy would require that any change in that location should have negligible effect on the published output, making it impossible to communicate any useful information to the service provider.

To overcome this issue, Dewri [2012] proposes a mix of differential privacy and $k$-anonymity, by fixing an anonymity set of $k$ locations and requiring that the probability to report the same obfuscated location $z$ from any of these $k$ locations should be similar (up to $\epsilon'$). This property is achieved by adding Laplace noise to each Cartesian coordinate independently. There are however two problems with this definition: first, the choice of the anonymity set crucially affects the resulting privacy; outside this set no privacy is guaranteed at all. Second, the property itself is rather weak; reporting the geometric median (or any deterministic function) of the $k$ locations would satisfy the same definition, although the privacy guarantee would be substantially lower than using Laplace noise.

Nevertheless, Dewri’s intuition of using Laplace noise$^1$ for location privacy is valid, and [Dewri 2012] provides extensive experimental analysis supporting this claim. Our notion of geo-indistinguishability provides the formal background for justifying the use of Laplace noise, while avoiding the need to fix an anonymity set by using the generalized variant of differential privacy developed by Chatzikokolakis et al..

Other location-privacy metrics.. [Cheng et al. 2006] proposes a location cloaking mechanism, and focuses on the evaluation of Location-based Range Queries. The degree of privacy is measured by the size of the cloak (also called uncertainty region), and by the coverage of sensitive regions, which is the ratio between the area of the cloak and the area of the regions inside the cloak that the user considers to be sensitive. In order to deal with the side-information that the attacker may have, ad-hoc solutions are proposed, like patching cloaks to enlarge the uncertainty region or delaying requests. Both solutions may cause a degradation in the quality of service.

Ardagna et al. [2007] observe that most of the times the real location of the user already has some level of inaccuracy, due to the specific sensing technology or to the environmental conditions. Different obfuscation techniques are then used to increase this inaccuracy in order to achieve a certain level of privacy. This level of privacy is defined as the ratio between the accuracy before and after the application of the obfuscation techniques.

Similar to the case of $k$-anonymity, both privacy metrics mentioned above make implicit assumptions about the adversary’s side information. This may imply a violation of the privacy definition in a scenario where the adversary has some knowledge about the user’s real location.

Transformation-based approaches.. A number of approaches for location privacy are radically different from the ones mentioned so far. Instead of cloaking the user’s location, they aim at making it completely invisible to the service provider. This is achieved by transforming all data to a different space, usually employing cryptographic techniques, so that they can be mapped back to spatial information only by the user.

$^1$The planar Laplace distribution that we use in our work, however, is different from the distribution obtained by adding Laplace noise to each Cartesian coordinate, and has better differential privacy properties (c.f. Section 4.1).
The data stored in the provider, as well as the location send by the user are encrypted. Then, using techniques from private information retrieval, the provider can return information about the encrypted location, without ever discovering which actual location it corresponds to.

A drawback of these techniques is that they are computationally demanding, making it difficult to implement them in a handheld device. Moreover, they require the provider's data to be encrypted, making it impossible to use existing providers, such as Google Maps, which have access to the real data.

Effectiveness of attacks. An indirect way of assessing the privacy guarantees of a mechanism is to measure the effectiveness of various location inference attacks. Several works present attacks and practical challenges for location privacy. Gambs et al. [2011] developed and tested a toolkit for inference attacks on the reported locations of users to discover points of interests, future locations and co-location of two individuals. The same technique was employed in [Gambs et al. 2013] focusing on de-anonymization attacks with the goal of evaluating the effectiveness of sanitization mechanisms. Primault et al. tested the resilience of Geo-Indistinguishability to identification of Points of Interests of users over two real GPS traces datasets, with varying level of privacy (and therefore noise).

3. GEO-INDISTINGUISHABILITY

In this section we formalize our notion of geo-indistinguishability. As already discussed in the introduction, the main idea behind this notion is that, for any radius \( r > 0 \), the user enjoys \( \varepsilon r \)-privacy within \( r \), i.e. the level of privacy is proportional to the radius. Note that the parameter \( \varepsilon \) corresponds to the level of privacy at one unit of distance. For the user, a simple way to specify his privacy requirements is by a tuple \((\varepsilon, r)\), where \( r \) is the radius he is mostly concerned with and \( \varepsilon \) is the privacy level he wishes for that radius. In this case, it is sufficient to require \( \varepsilon \)-geo-indistinguishability for \( \varepsilon = \ell / r \); this will ensure a level of privacy \( \ell \) within \( r \), and a proportionally selected level for all other radii.

So far we kept the discussion on an informal level by avoiding to explicitly define what \( \ell \)-privacy within \( r \) means. In the remaining of this section we give a formal definition, as well as two characterizations which clarify the privacy guarantees provided by geo-indistinguishability.

Probabilistic model. We first introduce a simple model used in the rest of the paper. We start with a set \( \mathcal{X} \) of points of interest, typically the user’s possible locations. Moreover, let \( \mathcal{Z} \) be a set of possible reported values, which in general can be arbitrary, allowing to report obfuscated locations, cloaking regions, sets of locations, etc. However, to simplify the discussion, we sometimes consider \( \mathcal{Z} \) to also contain spatial points, assuming an operational scenario of a user located at \( x \in \mathcal{X} \) and communicating to the attacker a randomly selected location \( z \in \mathcal{Z} \) (e.g. an obfuscated point).

Probabilities come into place in two ways. First, the attacker might have side information about the user’s location, knowing, for example, that he is likely to be visiting the Eiffel Tower, while unlikely to be swimming in the Seine river. The attacker’s side information can be modeled by a prior distribution \( \pi \) on \( \mathcal{X} \), where \( \pi(x) \) is the probability assigned to the location \( x \).

Second, the selection of a reported value in \( \mathcal{Z} \) is itself probabilistic; for instance, \( z \) can be obtained by adding random noise to the actual location \( x \) (a technique used in Section 4). A mechanism \( K \) is a probabilistic function for selecting a reported value; i.e. \( K \) is a function assigning to each location \( x \in \mathcal{X} \) a probability distribution on \( \mathcal{Z} \), where \( K(x)(\mathcal{Z}) \) is the probability that the reported point belongs to the set \( \mathcal{Z} \subseteq \mathcal{Z} \), when the
user’s location is $x$.\footnote{For simplicity we assume distributions on $\mathcal{X}$ to be discrete, but allow those on $\mathcal{Z}$ to be continuous (c.f. Section 4). All sets to which probability is assigned are implicitly assumed to be measurable.} Starting from $\pi$ and using Bayes’ rule, each observation $Z \subseteq \mathcal{Z}$ of a mechanism $K$ induces a \textit{posterior} distribution $\sigma = \text{Bayes}(\pi, K, Z)$ on $\mathcal{X}$, defined as $\sigma(x) = \frac{K(x)(Z)\pi(x)}{\sum_y K(y)(Z)\pi(y)}$.

We define the \textit{multiplicative distance} between two distributions $\sigma_1, \sigma_2$ on some set $\mathcal{S}$ as $d_P(\sigma_1, \sigma_2) = \sup_{\mathcal{S}} |\ln \frac{\sigma_1(S)}{\sigma_2(S)}|$, with the convention that $|\ln \frac{\sigma_1(S)}{\sigma_2(S)}| = 0$ if both $\sigma_1(S), \sigma_2(S)$ are zero and $\infty$ if only one of them is zero.

We are now ready to state our definition of geo-indistinguishability. Intuitively, a privacy requirement is a constraint on the distributions $K(x), K(x')$ produced by two different points $x, x'$. Let $d_2(\cdot, \cdot)$ denote the Euclidean metric. Enjoying $\ell$-privacy within $r$ means that for any $x, x'$ s.t. $d_2(x, x') \leq r$, the distance $d_P(K(x), K(x'))$ between the corresponding distributions should be at most $\ell$. Then, requiring $\epsilon \ell$-privacy for all radii $r$, forces the two distributions to be similar for locations close to each other, while relaxing the constraint for those far away from each other, allowing a service provider to distinguish points in Paris from those in London.

\textbf{Definition 3.1 (geo-indistinguishability).} A mechanism $K$ satisfies $\epsilon$-geo-indistinguishability iff for all $x, x'$:

$$d_P(K(x), K(x')) \leq \epsilon d_2(x, x')$$

Equivalently, the definition can be formulated as $K(x)(Z) \leq e^{\epsilon d_2(x, x')} K(x')(Z)$ for all $x, x' \in \mathcal{X}, Z \subseteq \mathcal{Z}$. Note that for all points $x'$ within a radius $r$ from $x$, the definition forces the corresponding distributions to be at most $\epsilon r$ distant.

The quantity $\epsilon d_2(x, x')$ can be viewed as the \textit{distinguishability level} between the secrets $x$ and $x'$. The use of the Euclidean metric $d_2$ is natural for location privacy: the closer (geographically) two points are, the less distinguishable we would like them to be. Note, however, that other metrics could be used instead of $d_2$, such as the Manhattan metric or driving distance, depending on the application. The definition that we obtain by using an arbitrary distinguishability metric $d_h$, i.e. requiring that $d_P(K(x), K(x')) \leq d_h(x, x')$, is referred to as $d_h$-privacy\footnote[5]{Note that we can generally consider the scaling factor $\epsilon$ to be part of the metric, although sometimes we emphasize it by talking of $\epsilon d_h$-privacy}, and is studied on its own right in \cite{Chatzikokolakis et al. 2013}. Some of the results of this paper do not depend on the actual metric, so they are given in the general framework of $d_h$-privacy.

Note also that standard differential privacy simply corresponds to $\epsilon d_h(x, x')$-privacy, where $d_h$ is the Hamming distance between databases $x, x'$, i.e. the number of individuals in which they differ. However, in our scenario, using the Hamming metric of standard differential privacy – which aims at completely protecting the value of an individual – would be too strong, since the only information is the location of a single individual. Nevertheless, we are not interested in completely hiding the user’s location, since some approximate information needs to be revealed in order to obtain the required service. Hence, using a privacy level that depends on the Euclidean distance between locations is a natural choice.

\textit{Protecting location traces.} So far, we have assumed a sporadic use of an LBS, meaning that the service is used infrequently enough that we can assume no correlation between different uses and treat each one of them independently. In this case, the user’s secret is a single location. In the case of repeated use, however, the user forms a location trace which should be protected; the provider is allowed to obtain only approximate information about the locations, their exact value should be kept private.
In this case, the secret is the trace, i.e. a tuple of points denoted by $x = [x_1, \ldots, x_n]$, while $x[i]$ denotes the $i$-th element of the trace. The notion of $\epsilon$-geo-indistinguishability extends naturally by defining the distance between two tuples $x, x'$ as:

$$d_\infty(x, x') = \max_i d_2(x[i], x'[i])$$

and using $\epsilon d_\infty$-privacy as our privacy definition. Following the idea of reasoning within a radius $r$, this definition requires that two traces at most $r$ away from each other (i.e. such that $x[i], x'[i]$ are all within distance $r$ from each other) should produce distributions at most $\epsilon r$ apart.

### 3.1. Characterizations

In this section we state two characterizations of geo-indistinguishability, obtained from the corresponding results of [Chatzikokolakis et al. 2013] (for general metrics), which provide intuitive interpretations of the privacy guarantees offered by this notion.

**Adversary’s conclusions under hiding.** The first characterization uses the concept of a hiding function $\phi : \mathcal{X} \rightarrow \mathcal{X}$. The idea is that $\phi$ can be applied to the user’s actual location before the mechanism $K$, so that the latter has only access to a hidden version $\phi(x)$, instead of the real location $x$. A mechanism $K$ with hiding applied is simply the composition $K \circ \phi$. Intuitively, a location remains private if, regardless of his side knowledge (captured by his prior distribution), an adversary draws the same conclusions (captured by his posterior distribution), regardless of whether hiding has been applied or not. However, if $\phi$ replaces locations in Paris with those in London, then clearly the adversary’s conclusions will be greatly affected. Hence, we require that the effect on the conclusions depends on the maximum distance $d_2(\phi) = \sup_{x \in \mathcal{X}} d_2(x, \phi(x))$ between the real and hidden location.

**Theorem 3.2.** A mechanism $K$ satisfies $\epsilon$-geo-indistinguishability iff for all $\phi : \mathcal{X} \rightarrow \mathcal{X}$, all priors $\pi$ on $\mathcal{X}$, and all $Z \subseteq \mathcal{Z}$:

$$d_p(\sigma_1, \sigma_2) \leq 2\epsilon d_2(\phi)$$

where $\sigma_1 = \text{Bayes}(\pi, K, Z)$ and $\sigma_2 = \text{Bayes}(\pi, K \circ \phi, Z)$

Note that this is a natural adaptation of a well-known interpretation of standard differential privacy, stating that the attacker’s conclusions are similar, regardless of his side knowledge, and regardless of whether an individual’s real value has been used in the query or not. This corresponds to a hiding function $\phi$ removing the value of an individual.

Note also that the above characterization compares two posterior distributions. Both $\sigma_1, \sigma_2$ can be substantially different than the initial knowledge $\pi$, which means that an adversary does learn some information about the user’s location.

**Knowledge of an informed attacker.** A different approach is to measure how much the adversary learns about the user’s location, by comparing his prior and posterior distributions. However, since some information is allowed to be revealed by design, these distributions can be far apart. Still, we can consider an informed adversary who already knows that the user is located within a set $N \subseteq \mathcal{X}$. Let $d_2(N) = \sup_{x, x' \in N} d_2(x, x')$ be the maximum distance between points in $x$. Intuitively, the user’s location remains private if, regardless of his prior knowledge within $N$, the knowledge obtained by such an informed adversary should be limited by a factor depending on $d_2(N)$. This means that if $d_2(N)$ is small, i.e. the adversary already knows the location with some accuracy, then the information that he obtains is also small, meaning that he cannot improve his accuracy. Denoting by $\pi|_N$ the distribution obtained from $\pi$ by restricting to $N$ (i.e. $\pi|_N(x) = \pi(x|N)$), we obtain the following characterization:
THEOREM 3.3. A mechanism $K$ satisfies $\epsilon$-geo-indistinguishability iff for all $N \subseteq X$, all priors $\pi$ on $X$, and all $Z \subseteq Z$:

$$d_P(\pi|N, \sigma|N) \leq \epsilon d_2(N)$$

where $\sigma = \text{Bayes}(\pi, K, Z)$

Note that this is a natural adaptation of a well-known interpretation of standard differential privacy, stating that an informed adversary who already knows all values except individual's $i$, gains no extra knowledge from the reported answer, regardless of side knowledge about $i$'s value [Dwork et al. 2006].

Abstracting from side information. A major difference of geo-indistinguishability, compared to similar approaches from the literature, is that it abstracts from the side information available to the adversary, i.e. from the prior distribution. This is a subtle issue, and often a source of confusion, thus we would like to clarify what “abstracting from the prior” means. The goal of a privacy definition is to restrict the information leakage caused by the observation. Note that the lack of leakage does not mean that the user's location cannot be inferred (it could be inferred by the prior alone), but instead that the adversary's knowledge does not increase due to the observation.

However, in the context of LBSs, no privacy definition can ensure a small leakage under any prior, and at the same time allow reasonable utility. Consider, for instance, an attacker who knows that the user is located at some airport, but not which one. The attacker's prior knowledge is very limited, still any useful LBS query should reveal at least the user's city, from which the exact location (i.e. the city's airport) can be inferred. Clearly, due to the side information, the leakage caused by the observation is high.

So, since we cannot eliminate leakage under any prior, how can we give a reasonable privacy definition without restricting to a particular one? First, we give a formulation (Definition 3.1) which does not involve the prior at all, allowing to verify it without knowing the prior. At the same time, we give two characterizations which explicitly quantify over all priors, shedding light on how the prior affects the privacy guarantees.

4. MECHANISMS FOR THE SPORADIC CASE

In this section we present two mechanisms for applying noise to a single location while satisfying geo-indistinguishability. The first one, the planar Laplace mechanism, is a simple and efficient mechanism that scales to any number of possible locations while being generic and independent from the user's behaviour. The second is adapted to a specific user and guarantees optimal utility (or minimum quality loss) for that user, however it is only applicable when the number of possible locations is limited.

4.1. The planar Laplace mechanism

We start by defining a mechanism for geo-indistinguishability on the continuous plane. The idea is that whenever the actual location is $x \in \mathbb{R}^2$, we report, instead, a point $z \in \mathbb{R}^2$ generated randomly according to a distribution with probability density function:

$$D_\epsilon(z) = \frac{e^{\frac{-\epsilon}{2} d_2(x,z)}}{2\pi}$$

This function is called the planar Laplace centered at $x$ and is illustrated in Figure 1. The resulting mechanism can be shown to satisfy $\epsilon$-geo-indistinguishability [Andrés et al. 2013].

Note that this definition of the two-dimensional Laplace distribution follows [Lange and Sinsheimer 1993] and is different than generating the two coordinates independently from a standard (one dimensional) Laplace distribution. Such an approach would not, in fact, satisfy geo-indistinguishability.
Drawing a random point. We illustrate now how to draw a random point from the pdf defined in (1). First of all, we note that the pdf of the planar Laplace distribution depends only on the distance from $x$. It will be convenient, therefore, to switch to a system of polar coordinates with origin $x$. A point $z$ will be represented as a point $(r, \theta)$, where $r$ is the distance of $z$ from $x$, and $\theta$ is the angle that the line $xz$ forms with respect to the horizontal axis of the Cartesian system. After the transformation, the pdf of the polar Laplace centered at the origin $x$ is:

$$D_\epsilon(r, \theta) = \frac{\epsilon^2}{2\pi} r e^{-\epsilon r}$$

Let $R, \Theta$ be the random variables representing the radius and the angle; the property that allows to efficiently draw from the polar Laplace is that the two variables are independent, that is $D_\epsilon(r, \theta)$ is the product of the two marginals:

$$D_{\epsilon,R}(r) = \int_0^{2\pi} D_\epsilon(r, \theta) d\theta = \frac{\epsilon^2}{2\pi} r e^{-\epsilon r}$$

$$D_{\epsilon,\Theta}(\theta) = \int_0^\infty D_\epsilon(r, \theta) dr = \frac{1}{2\pi}$$

Note that $D_{\epsilon,R}(r)$ corresponds to the gamma distribution with shape 2 and scale $1/\epsilon$.

Hence, in order to draw a point $(r, \theta)$ it is sufficient to draw separately $r$ and $\theta$ from $D_{\epsilon,R}(r)$ and $D_{\epsilon,\Theta}(\theta)$ respectively. Since $D_{\epsilon,\Theta}(\theta)$ is constant, $\theta$ can be drawn from a uniform distribution on the interval $[0, 2\pi)$.

We now show how to draw $r$. Following standard lines, we consider the cumulative distribution function (cdf) $C_\epsilon(r)$:

$$C_\epsilon(r) = \int_0^r D_{\epsilon,R}(\rho) d\rho = 1 - (1 + \epsilon r) e^{-\epsilon r}$$

Intuitively, $C_\epsilon(r)$ represents the probability that the radius of the random point falls between $0$ and $r$. Finally, we generate a random number $\rho$ with uniform probability in the interval $[0, 1)$, and we set $r = C_\epsilon^{-1}(\rho)$. Note that

$$C_\epsilon^{-1}(\rho) = -\frac{1}{\epsilon} (W_{-1}(-e^{-1}) + 1)$$

where $W_{-1}$ is the Lambert W function (the $-1$ branch), which can be computed efficiently and is implemented in several numerical libraries.

Note that in practice only a discretized version of the continuous mechanism can be implemented; the discretized variant can be shown to also satisfy geo-indistinguishability, for a slightly bigger $\epsilon$, although the difference is negligible on a double precision machine. A detailed discussion of discretization issues can be found in [Andrés et al. 2013].

The planar Laplace mechanism has two main advantages: first, it is simple and efficient to compute without restricting the number of possible locations. Second, it can be applied to a generic user without prior information on his behaviour. The usefulness of the mechanism for generic applications is showcased in Location Guard [https://github.com/chatziko/location-guard], a browser extension for Chrome and Firefox, which provides location privacy for websites accessing the user’s location through
the HTML5 geolocation API, by adding noise to the reported location using the planar Laplace mechanism.

On the other hand, being generic, the planar Laplace mechanism offers no optimality guarantees for the quality loss of the reported location. In the following section, we show how to improve utility by construct mechanisms adapted to the behaviour of a particular user.

4.2. Geo-indistinguishable mechanisms of optimal utility

The goal of a privacy mechanism is not to hide completely the secret but to disclose enough information to be useful for some service while hiding the rest to protect the user's privacy. Typically these two requirements go in opposite directions: a stronger privacy level requires more noise which results in a lower utility.

From the user's point of view, we want to quantify the service quality loss (QL) produced by the mechanism $K$. Given a quality metric $d_Q$ on locations, such that $d_Q(x, z)$ measures how much the quality decreases by reporting $z$ when the real location is $x$ (the Euclidean metric $d_2$ being a typical choice), we can naturally define the quality loss as the expected distance between the real and the reported location, that is

$$QL(K, \pi, d_Q) = \sum_{x,z} \pi(x)K(x)(z)d_Q(x, z)$$

where $\pi$ is a prior on $X$ modeling the user's behaviour.

Despite the generality of the planar Laplace mechanism, in some cases we want to be able to build a mechanism that optimizes the trade-off between privacy (in terms of geo-indistinguishability) and quality loss (in terms of QL) for a specific user. Our main goal is, given a set of locations $X$ with a privacy metric $d_X$, a privacy level $\epsilon$, a user profile $\pi$ and a quality metric $d_Q$, to find an $\epsilon d_X$-private mechanism such that its QL is as small as possible. We start by describing a set of linear constraints that enforce $\epsilon d_X$-privacy, which allows to obtain an optimal mechanism as a linear optimization problem. However, the number of constraints can be large, making the approach computationally demanding as the number of locations increases. As a consequence, we then propose an approximate solution that replaces $d_X$ with the metric induced by a spanning graph.

Constructing an optimal mechanism. The constructed mechanism is assumed to have as both input and output a predetermined finite set of locations $X$. For instance, $X$ can be constructed by dividing the map in a finite number of regions (of arbitrary size and shape), and selecting in $X$ a representative location for each region. We also assume a prior $\pi$ over $X$, representing the probability of the user being at each location at any given time. Since $X$ is finite, a mechanism $K$ can be represented by a stochastic matrix, where $k_{xz}$ is the probability to report $z$ from location $x$.

Given a privacy metric $d_X$ and a privacy parameter $\epsilon$, the goal is to construct a $\epsilon d_X$-private mechanism $K$ such that the service quality loss with respect to a quality metric $d_Q$ is minimum. This property is formally defined below:

**Definition 4.1.** Given a prior $\pi$, a privacy metric $d_X$, a privacy parameter $\epsilon$ and a quality metric $d_Q$, a mechanism $K$ is $\epsilon d_X$-OPTQL($\pi, d_Q$) iff:

1. $K$ is $\epsilon d_X$-private, and
2. for all mechanisms $K'$, if $K'$ is $\epsilon d_X$-private then $QL(K, \pi, d_Q) \leq QL(K', \pi, d_Q)$

In order for $K$ to be $\epsilon d_X$-private it should satisfy the following constraints:

$$k_{xz} \leq e^{\epsilon d_X(x, x')}k_{x'z} \quad x, x', z \in X$$
Hence, we can construct an optimal mechanism by solving a linear optimization problem, minimizing $Q_L(K, \pi, d_Q)$ while satisfying $\epsilon d_x$-privacy:

Minimize: $\sum_{x, z \in X} \pi_x k_{xz} d_Q(x, z)$

Subject to: $k_{x'z} \leq e^{\epsilon d_x(x, x')} k_{x'z}$  \hspace{1cm} x, x', z \in X
$\sum_{z \in X} k_{xz} = 1$  \hspace{1cm} x \in X
$k_{xz} \geq 0$  \hspace{1cm} x, z \in X

It is easy to see that the mechanism $K$ generated by the previous optimization problem is $\epsilon d_x$-OPTQL($\pi, d_Q$).

A more efficient method using spanners. In the optimization problem of the previous section, the $\epsilon d_x$-privacy definition introduces $|X|^3$ constraints in the linear program. However, in order to be able to manage a large number of locations, we would like to reduce this amount to a number in the order of $O(|X|^2)$.

So far we are not making any assumption about $d_x$, and therefore we need to specify $|X|$ constraints for each pair of locations $x$ and $x'$. However, it is worth noting that if the distance $d_x$ is induced by a weighted graph (i.e. the distance between each pair of locations is the weight of a minimum path in a graph), then we only need to consider $|X|$ constraints for each pair of locations that are adjacent in the graph.

It might be the case, though, that the metric $d_x$ is not induced by any graph (other than the complete graph), and consequently the amount of constraints remains the same. In fact, this is generally the case for the Euclidean metric. Therefore, we consider the case in which $d_x$ can be approximated by some graph-induced metric.

If $G$ is an undirected weighted graph, we denote with $d_G$ the distance function induced by $G$, i.e. $d_G(x, x')$ denotes the weight of a minimum path between the nodes $x$ and $x'$ in $G$. Then, if the set of nodes of $G$ is $X$ and the weight of its edges is given by the metric $d_x$, we can approximate $d_x$ with $d_G$. In this case, we say that $G$ is a spanning graph, or a spanner [Narasimhan and Smid 2007; Sack and Urrutia 1999], of $X$.

**Definition 4.2 (Spanner).** A weighted graph $G = (X, E)$, with $E \subseteq X \times X$ and weight function $w : E \rightarrow \mathbb{R}$ is a spanner of $X$ if

$$w(x, x') = d_x(x, x') \hspace{1cm} \forall (x, x') \in E$$

Note that if $G$ is a spanner of $X$, then

$$d_G(x, x') \geq d_x(x, x') \hspace{1cm} \forall x, x' \in X$$

A main concept in the theory of spanners is that of dilation, also known as stretch factor:

**Definition 4.3 (Dilation).** Let $G = (X, E)$ be a spanner of $X$. The dilation of $G$ is calculated as:

$$\delta = \max_{x \neq x' \in X} \frac{d_G(x, x')}{d_x(x, x')}$$

A spanner of $X$ with dilation $\delta$ is called a $\delta$-spanner of $X$.

Informally, a $\delta$-spanner of $X$ can be considered an approximation of the metric $d_x$ in which distances between nodes are “stretched” by a factor of at most $\delta$.

If $G$ is a $\delta$-spanner of $X$, then it holds that

$$d_G(x, x') \leq \delta d_x(x, x') \hspace{1cm} \forall x, x' \in X$$
which leads to the following proposition:

**Proposition 4.4.** Let \( X \) be a set of locations with metric \( d_X \), and let \( G \) be a \( \delta \)-spanner of \( X \). If a mechanism \( K \) for \( X \) is \( \frac{\delta}{5}d_G \)-private, then \( K \) is \( \epsilon d_X \)-private.

We can then propose a new optimization problem to obtain a \( \epsilon d_X \)-private mechanism. If \( G = (X, E) \) is a \( \delta \)-spanner of \( X \), we require not the constraints corresponding to \( \epsilon d_X \)-privacy, but those corresponding to \( \frac{\delta}{5}d_G \)-privacy instead, that is, \( |X| \) constraints for each edge of \( G \):

\[
\text{Minimize: } \sum_{x, z \in X} \pi_x k_{xz} d_Q(x, z) \\
\text{Subject to: } k_{xz} \leq e^{\frac{\delta}{5}d_G(x, x')} k_{x'z} \quad z \in X, (x, x') \in E \\
\sum_{x \in X} k_{xz} = 1 \quad x \in X \\
k_{xz} \geq 0 \quad x, z \in X
\]

Since the resulting mechanism is \( \frac{\delta}{5}d_G \)-private, by Proposition 4.4 it must also be \( \epsilon d_X \)-private. However, the number of constraints induced by \( \frac{\delta}{5}d_G \)-privacy is now \( |E||X| \). Moreover, as discussed in the next section, for any \( \delta > 1 \) there is an algorithm that generates a \( \delta \)-spanner with \( O(\frac{|X|^2}{\delta}) \) edges, which means that, fixing \( \delta \), the total number of constraints of the linear program is \( O(|X|^2) \).

It is worth noting that although \( \epsilon d_X \)-privacy is guaranteed, optimality is lost: the obtained mechanism is \( \frac{\delta}{5}d_G \)-OPTQL(\( \pi, d_Q \)) but not necessarily \( \epsilon d_X \)-OPTQL(\( \pi, d_Q \)), since the set of \( \frac{\delta}{5}d_G \)-private mechanisms is a subset of the set of \( \epsilon d_X \)-private mechanisms. The QL of the obtained mechanism will now depend on the dilation \( \delta \) of the spanner: the smaller \( \delta \) is, the closer the QL of the mechanism will be from the optimal one. In consequence, there is a trade-off between the accuracy of the approximation and the number of constraints in linear program.

**5. Mechanisms for the Repeated Case**

In the previous section we considered a sporadic use of a service, in which case only a single location needs to be obfuscated. We now turn our attention to the repeated case, in which the user’s location trace (sometimes called trajectory in the literature) needs to be protected. We denote by \( x = [x_1, \ldots, x_n] \) a trace, by \( x[i] \) the \( i \)-th element of \( x \), by \([\cdot]\) the empty trace and by \( x::x \) the trace obtained by adding \( x \) to the head of \( x \). We also define \( \text{tail}(x::x) = x \). As already discussed in Section 3, geo-indistinguishability can be naturally extended to the case of location traces by using \( d_\infty \) as the underlying distinguishability metric.

**mechanism** IM(x)

\[
z := [\cdot] \\
\text{for } i := 1 \text{ to } |x| \\
z := N(\epsilon_N)(x[i]) \\
z := z::z \\
\text{return } z
\]

Fig. 2: Independent Mechanism

increases linearly with \( n \).

5.0.1. Independent Mechanism. In order to sanitize \( x \) we can simply apply a noise mechanism independently to each secret \( x_i \). We assume that a family of noise mechanisms \( N(\epsilon_N) : X \rightarrow \mathcal{P}(Z) \) are available, parametrized by \( \epsilon_N \), where each mechanism \( N(\epsilon_N) \) satisfies \( \epsilon_N \)-privacy. Both mechanisms of Section 4 can be used for this purpose. The resulting mechanism, called the independent mechanism \( IM : X^n \rightarrow \mathcal{P}(Z^n) \), is shown in Figure 2. As explained in the introduction, the main issue with IM is that it is \( n\epsilon d_\infty \)-private, i.e. the budget consumed in-
5.0.2. A predictive $d_\epsilon$-private mechanism. We introduce now our prediction-based approach. The fundamental intuition is that the correlation of the points in the trace can be exploited to the advantage of the mechanism. A simple way of doing this is to try to predict new points from past information; if the point can be predicted with enough accuracy it is called easy; in this case the prediction can be reported without adding new noise. One the other hand, hard points, that is those that cannot be predicted, are sanitized with new noise. However testing if a point is easy or hard reveals some information about the real location and violates $d_\epsilon$-privacy as for different locations we might have different answers. In order to respect the definition we will need to make the test $d_\epsilon$-private itself, reducing its precision and adding a new cost to our global budget. We will show that with enough correlation in the input the gain in predicted points is worth the cost of the test.

Let $B = \{0, 1\}$. A boolean $b \in B$ denotes whether a point is easy (0) or hard (1). A sequence $r = [z_1, b_1, \ldots, z_n, b_n]$ of reported values and booleans is called a run; the set of all runs is denoted by $\mathcal{R} = (\mathbb{Z} \times B)^*$. A run will be the output of our predictive mechanism; note that the booleans $b_i$ are considered public and will be reported by the mechanism.

Main components. The predictive mechanism has three main components: first, the prediction is a deterministic function $\Omega : \mathcal{R} \rightarrow \mathbb{Z}$, taking as input the run reported up to this moment and trying to predict the next reported point, which should be at an acceptable distance from the actual one. The output of the prediction function is denoted by $\tilde{z} = \Omega(r)$. Note that the possibility of a successful prediction should not be viewed as a privacy violation because $\Omega$ predicts the reported location, not the actual one.

Second, a test is a family of mechanisms $\Theta(\epsilon_\theta, l, \tilde{z}) : \mathcal{X} \rightarrow \mathcal{P}(B)$, parametrized by $\epsilon_\theta, l, \tilde{z}$. The test takes as input the point $x$ and reports whether the prediction $\tilde{z}$ is acceptable or not for this point. If the test is successful then the prediction will be used instead of generating new noise. The purpose of the test is to guarantee a certain level of utility: predictions that are farther than the threshold $l$ should be rejected. Since the test is accessing the actual location, it should be private itself, where $\epsilon_\theta$ is the allowed budget for testing.

The test mechanism that will be used throughout the paper is the one below, which is based on adding Laplace noise to the threshold $l$:

$$\Theta(\epsilon_\theta, l, \tilde{z})(x) = \begin{cases} 0 & \text{if } d_x(x, \tilde{z}) \leq l + \text{Lap}(\epsilon_\theta) \\ 1 & \text{ow.} \end{cases} \quad (3)$$

The test is defined for all $\epsilon_\theta > 0, l \in [0, +\infty), \tilde{z} \in \mathbb{Z}$, and can be used for any metric $d_x$, as long as the domain of reported locations is the same as the one of the actual locations, so that $d_x(x, \tilde{z})$ is well defined.

Finally, a noise mechanism is a family of mechanisms $N(\epsilon_N) : \mathcal{X} \rightarrow \mathcal{P}(\mathbb{Z})$, parametrized by the available budget $\epsilon_N$. The noise mechanism is used for hard secrets that cannot be predicted and can be any of the sporadic mechanisms presented in Section 4, although in the following we will assume the use of the planar Laplace for simplicity.

Budget management. The parameters of the mechanism’s components need to be configured at each step. This can be done in a dynamic way using the concept of a budget manager. A budget manager $\beta$ is a function that takes as input the run produced so far and returns the budget and the threshold to be used for the test at this step as well as the budget for the noise mechanism: $\beta(r) = (\epsilon_\theta, \epsilon_N, l)$.

Of course the amount of budget used for the test should always be less than the amount devoted to the noise, otherwise it would be more convenient to just use the
mechanism PM(x)

\[ r := [] \]
\[ \text{for } i := 1 \text{ to } |x| \]
\[ (z, b) := \text{Step}(r)(x[i]) \]
\[ r := (z, b) :: r \]
\[ \text{return } r \]

 mechanism Step(r)(x)

\[ (\epsilon_\beta, \epsilon_N, l) := \beta(r) \]
\[ \tilde{z} := \Omega(r) \]
\[ b := \Theta(\epsilon_\beta, l, \tilde{z})(x) \]
\[ \text{if } b == 0 \text{ then } z := \tilde{z} \]
\[ \text{else } z := N(\epsilon_N)(x) \]
\[ \text{return } (z, b) \]

(a) Predictive Mechanism  

(b) Single step of the Predictive Mechanism

Fig. 3: The mechanisms

independent noise mechanism. Still, there is great flexibility in configuring the various parameters and several strategies can be implemented in terms of a budget manager.

The mechanism. We are now ready to fully describe our mechanism. A single step of the predictive mechanism, displayed in Figure 3b, is a family of mechanisms \( \text{Step}(r) : X \to \mathcal{P}(Z \times B) \), parametrized by the run \( r \) reported up to this point. The mechanism takes a location \( x \) and returns a reported location \( z \), as well as a boolean \( b \) denoting whether the secret was easy or hard. First, the mechanism obtains the various configuration parameters from the budget manager as well as a prediction \( \tilde{z} \). Then the prediction is tested using the test mechanism. If the test is successful the prediction is returned, otherwise a new reported location is generated using the noise mechanism.

Finally, the predictive mechanism, displayed in Figure 3a, is a mechanism \( \text{PM} : X^n \to \mathcal{P}(R) \). It takes as input a trace \( x \), and applies \( \text{Step}(r) \) to each point, while extending at each step the run \( r \) with the new reported values \((z, b)\).

Note that an important advantage of the mechanism is that it is online, that is the sanitization of each location does not depend on future ones. This means that the user can query at any time during the life of the system, as opposed to offline mechanisms where all the requests need to be generated before the sanitization.

The main innovation of this mechanism if the use of the prediction function, which allows to decouple the privacy mechanism from the correlation analysis, creating a family of modular mechanisms where by plugging in different predictions we are able to work in new domains.

5.0.3. Privacy. It can be shown that the predictive mechanism, given a family of test functions and noise functions respectively \( \epsilon_\beta \) and \( \epsilon_N \) \( d_X \)-private, is itself \( d_X \)-private. The global budget \( \epsilon_\beta(r) \) is actually dependent on the budget manager and on the specific run, which is incompatible with \( d_X \)-privacy that is always independent from the prior. The reason is that a hard step is more expensive than an easy step because of the cost of the noise mechanism. Therefore there is a difference between the budget spent on a “good” run, where the input has a considerable correlation, the prediction performs well and the majority of steps are easy, and a run with uncorrelated secrets, where any prediction is useless and all the steps are hard. In the latter case it is clear that our mechanism wastes part of its budget on tests that always fail, performing worse than an independent mechanism.

However we can still enforce the definition with the use of a \( \epsilon \)-bounded budget manager. Such a budget manager provides a fixed privacy guarantee by sacrificing utility: in the case of a bad run it either needs to lower the budget spend per secret, leading to more noise, or to stop early, handling a smaller number of requests. In this case the budget manager moves the impact of the runs away from the privacy budget and to utility. Two such managers were developed, both with fixed global privacy, one improv-
(a) Left: Quality loss of the O\textsubscript{PT}QL and PL mechanisms for different values of $\epsilon$. The mechanisms were calculated for all users. Here, points represent the utility for every user, while the two lines join the medians for each mechanism and each value of $\epsilon$.

(b) Right: Relation between the approximation ratio and the number of constraints in the linear program. This number is independent from the user and from the value of $\epsilon$.

Fig. 4

6. EVALUATION

We experimentally verify the effectiveness of our mechanisms on the motivating example of a user performing various activities in a city, using two large data sets of GPS trajectories in the Beijing urban area [Zheng et al. 2010; Yuan et al. 2010]. Geolife [Zheng et al. 2010] collects the movements of several users, using a variety of transportation means, including walking, while in Tdrive [Yuan et al. 2010] we find exclusively taxi drivers trajectories. Due to space restrictions, only a small part of the results are given here; a detailed evaluation is available in [Andrés et al. 2013; Chatzikokolakis et al. 2014; Bordenabe et al. 2014].

Optimal mechanism. To show the benefits of using a mechanism with optimal utility, we compare now the QL of the optimal mechanism (O\textsubscript{PT}QL) and of the planar Laplace (PL) when both are generated with the same privacy level $\epsilon$. We can see the results in Figure 4a. The O\textsubscript{PT}QL mechanism clearly offers a better utility to the user, while guaranteeing the same level of geo-indistinguishability.

Regarding the spanner approximation of the optimal mechanism, the relation between the dilation and the number of constraints is shown in Figure 4b. It is clear that the number of constraints decreases exponentially with respect to the dilation, and therefore even for small dilations (which in turn mean good approximations) the number of constraints is significantly reduced with the proposed approximation technique. For instance, we have 87250 constraints for $\delta = 1$ (the optimal case), and 25551 constraints for $\delta = 1.05$. This represents a decrease of 71% with respect to the optimal case, with only 1.05 approximation ratio.

Predictive mechanism. In order to model both frequent (easier to predict) as well as seldom users, the GPS traces were sampled with a different probability of jumping, i.e., performing a query with a long delay (one hour) after the previous one. The test included two budget managers, one optimizing QL for a fixed number of queries...
(a) Average and 90th percentile error for fixed-rate. 

(b) Budget consumption rate for fixed-ql.

Fig. 5: Predictive mechanism evaluation with two budget managers

(fixed-rate), the other reducing budget consumption to prolong the use of the system at a fixed QL (fixed-ql). The results, shown in Figure 5, show considerable improvements with respect to independently applied noise, for both managers: we are able to decrease the average error up to 40% and the budget consumption rate up to 64%. The improvements are significant enough to broaden the applicability of geo-indistinguishability to cases impossible before: in our experiments we cover 30 queries with reasonable error which is enough for a full day of usage; alternatively we can drive the error down from 5 km to 3 km, which make it acceptable for a variety of applications.

7. LOCATION GUARD
Location Guard is an open source [https://github.com/chatziko/location-guard] web browser extension based on geo-indistinguishability, that provides location privacy when using the HTML5 geolocation API. The extension has reached considerable popularity since its release, covering Chrome, Firefox and Opera browsers, and more recently moving to mobile devices with Firefox for Android.

7.0.1. A web browser extension. The Web browser is by far the most used interface between users and privacy sensitive services nowadays; its growth in popularity as a platform to develop applications makes it an ideal target for a privacy preserving tool. Browser extensions, allowed to run code with limited privileges in order to offer new functionalities – e.g. ad-blockers, search-engines – are becoming increasingly popular. For our purposes it is possible to intercept calls to the Geolocation API, sanitize the original location provided by the browser, then return a private version to the call-
ing application in a transparent way. A browser extensions allows to incorporate pri-

vacy in a great number of services, in a way that is familiar and easy to install.

7.0.2 Desktop and mobile. Despite the fact that location privacy is considered espe-
cially important in mobile devices, desktop users have reasons to be concerned as well. 
Their awareness of the problem is shown by the popularity of the first versions of Lo-
cation Guard, that were limited to desktop browsers.

Since release 1.2.0 (February 2015) Location Guard runs on Firefox for Android, 
currently the only mobile browser supporting extensions, and its mobile user base 
has been growing rapidly. Supporting mobile devices is crucial since they typically 
follow all users’ movements, while mobile-optimized websites ask for user’s location 
increasingly often.

Although on smartphones native apps are the most popular way to interact with 
online services, the growing popularity of web applications, to contrast the rampant 
fragmentation of mobile development, promise a larger coverage of services for Loca-
tion Guard in the near future. Furthermore Mozilla announced a new port of Firefox 
to iOS in the next year thus covering the other half of the mobile space.

7.0.3 Operation. Every web application runs in a separate environment and can ac-
access privileged information, such as the user’s location, through a JavaScript API 
provided by the browser. A browser extension has the ability to run JavaScript 
code with higher privileges than a normal page and, among other things, to mod-
ify the content of any web page. When a page is loaded and before any other code 
is executed, Location Guard injects a small snippet of JavaScript that redefines 
geolocation.getCurrentPosition, the main function provided by the Geolocation API 
to retrieve the current position. When the rest of the page code runs and tries to access 
this function, it gets intercepted by Location Guard, which in turn obtains the real 
location from the browser, sanitizes it and returns it to the page.

The location is sanitized through the use of random noise drawn from a Planar 
Laplace distribution. The amount of noise added can be configured easily with a 
single parameter, the privacy level. Location guard provides three predefined levels 
(high,medium,low) and the user is also free to pick any other value. Additionally the 
privacy level can be adjusted per domain, so that different protection can be applied 
to different services: a larger amount of noise can be added to a weather service as 
opposed to a point of interest search engine.

An advantage of geo-indistinguishability is that it is relatively intuitive to explain 
to the user the effect of changing the levels on privacy and utility. For a certain privacy 
level we can compute two radiuses \( r_p \) and \( r_u \), respectively the radius of privacy protection 
and of utility. \( r_p \) is the area of locations highly indistinguishable from the actual one, i.e. all locations producing the same sanitized one with similar probabilities. \( r_u \) is 
the area in which the reported location lies with high probability, thus giving an idea 
of the utility that the user can expect. Both these radiuses can be easily plotted on a 
map to give the user a direct impression of privacy and utility, according to the level of 
protection chosen.

Apart from sanitizing the real location, Location Guard supports reporting a fixed 
predefined location, which offers perfect privacy at the cost of very low utility.

7.0.4 Adoption. As of April 2015 Location Guard counts 8,849 active users in Google 
Chrome, 7,084 in Mozilla Firefox (including Android) and 3,086 downloads in Opera. 
Adoption has been mainly through the browser extension stores, as well as through 
technology blogs covering Location Guard [Brinkmann 2014; Korben 2015].
8. ONGOING WORK

One shortcoming of standard geo-indistinguishability is that the privacy level has to be fixed independently of the user location. So for example once set to have a protection in a radius of 200m, that is sufficient in a dense urban environment, the same protection will be provided when the user moves outside the city, possibly in sparsely populated area. The problem is described in more depth in [Chatzikokolakis et al. 2015], where a solution is proposed in the form of an elastic mechanism that adapts to semantic characteristics of each location, such as population and presence of POIs. However, the extreme flexibility of this mechanism, that can change its behavior for locations just 100 meters apart, comes with the cost of a heavy phase of pre-processing to build its semantic map, which is not suitable for Location Guard.

What we would need is a simplified version of the elastic mechanism where the noise level is adapted to large areas, small enough to distinguish a park from a residential area, but still easily computable. In order to build this set of tiles, Location Guard can query a number of online geographical services, to obtain a set of geographical polygons together with a quantitative description of the amount of privacy they provide (an equivalent of the privacy mass used in the elastic mechanism). This dataset should cover an area large enough to contain most of the user usual movement and it can easily reach a few tens of kilometers while retaining a small size. Once this small dataset is build, we have a mapping from polygons to their privacy mass. We can now use it to define a function \( l \) that for each location, finds the containing polygon and returns a privacy level adapted to the privacy mass provided by the polygon.

The mechanism described above, despite achieving the flexible behavior we needed, does not satisfy geo-indistinguishability. It is enough to notice that the level of protection, a public information of the mechanism, depends on the current location of the user, which is sensitive. In order to solve this problem we need to make \( l \) itself differentially private; a simple way to do it is to first sanitize the current location with a fixed privacy level and then feed it to \( l \). Post processing a sanitized location does not pose any threat to privacy and would allow the mechanism to reduce sharply the amount of noise added to location in very private area.

We are currently evaluating the use of two online services to build the dataset. In order to extract the geographic polygons of administrative areas and natural features (such as lakes and rivers) we are using OverPass Turbo (overpass-turbo.eu), that provides a simple API to query the OpenStreetMap database. Once obtained the polygons we use again Overpass Turbo to query for POIs densities and DBpedia [dbpedia.org] to obtain population densities from Wikipedia; this values are combined in a privacy mass measure of each polygon. Preliminary results are shown in Figure 8.

9. RELATED WORK

Several related works have been already presented in Section 2, a few more are discussed in this section.
On the side of the optimal mechanism construction, the work closest to ours is [Shokri 2014], which independently proposes a linear programming technique to construct an optimal obfuscation mechanism wrt either the expected adversary error or geo-indistinguishability. Although there is an overlap in the main construction (the optimization problem of Section 4.2), most of the results are substantially different. The approximation technique of Shokri [2014] consists of discarding some of the geo-indistinguishability constraints when the distance involved is larger than a certain lower bound. This affects the geo-indistinguishability guarantees of the mechanism, although the effect can be tuned by properly selecting the bound for discarding constraints. On the other hand, our approximation technique, based on spanning graphs, can be used to reduce the number of constraints from cubic to quadratic without jeopardizing the privacy guarantees, by accepting a small decrease on the utility.

On the side of the predictive mechanism, our work was mainly inspired by the median mechanism [Roth and Roughgarden 2010], a work on differential privacy for databases based on the idea of exploiting the correlation on the queries to improve the budget usage. The mechanism uses a concept similar to our prediction to determine the answer to the next query using only past answers. An analogous work is the multiplicative weights mechanism [Hardt and Rothblum 2010], again in the context of statistical databases. The mechanism keeps a parallel version of the database which is used to predict the next answer and in case of failure it is updated with a multiplicative weights technique.

A key difference from our context is that in the above works, several queries are performed against the same database. In our setting, however, the secret (the position of the user) is always changing, which requires to exploit correlations in the data. This scenario was explored also by Dwork et al., who considered the case of an evolving secret and developed a differentially private counter.

Another work very close in spirit to ours is [Merrill et al. 2013]. The authors of this paper also considered the problem of location privacy for location based services, and used random noise to conceal the actual location. However their work is mainly focused on exploiting the features of existing technology, and does not attempt to give a rigorous definition of privacy guarantees.

Recently, Fawaz and Shin [2014] proposed the Location Privacy Guardian, which is perhaps the most complete framework, in the current state of the art, for privacy protection within smartphone applications. They considered several potential sources
of privacy breaches (profiling, tracking, etc.) and proposed solutions for each of them. For location privacy, they use our Laplace mechanism.

10. CONCLUSION

In this paper we have presented a framework for achieving privacy in location-based applications, taking into account the desired level of protection as well as the side-information that the attacker might have about the user. The core of our proposal is a new notion of location privacy, that we call geo-indistinguishability. In order to ensure this kind of privacy protection in location-based services, we have proposed mechanisms that achieve geo-indistinguishability by perturbing the actual location with random noise. We have considered two kinds of mechanisms: the first one is universal, i.e., it does not depend on the user, and uses a bivariate version of the Laplace function as the density function of the noise. The second one is designed assuming a particular user, and for that user it achieves the optimal trade off between privacy and utility. This is done by formulating the optimal trade off as a linear programming problem, whose solution are the conditional probabilities that compose the noise matrix. Finally, we have considered the problem of traces, namely the repeated use of the mechanism to generate a sequence of points (a situation that may arise, for instance, when the user makes several requests to the service during a walk), and we have addressed the problem of the degradation of the level of privacy due to the correlation of the actual locations. We have proposed a method that limits the degradation by applying a prediction mechanism, which allows to generate new reported locations without applying the mechanism at each step. Finally, we have evaluated our methods and showed that they are a considerable improvement w.r.t. the state of the art, and that our proposal to limit the negative effects of the correlation in traces is effective in practice.

Acknowledgments. This work was partially supported by the MSR-INRIA joint lab, by the European Union 7th FP project MEALS, by the project ANR-12-IS02-001 PACE, and by the INRIA Large Scale Initiative CAPPRIS.

REFERENCES


It's a special pleasure to introduce this quarter's Semantics Column. The author is Prakash Panangaden, a longtime friend and leader in the theoretical community who needs no introduction to the SIGLOG community. His hard work was fundamental for the creation of SIGLOG, and his research has consistently attracted attention in a broad range of topics. This month's column is on one of those themes – probabilistic bisimulation. Prakash was the leader of research on labelled Markov processes which led to many of the results he reports in the column.

Probability theory first emerged in theoretical computer science when Michael Rabin introduced probabilistic automata in the early 1960s [4]. These automata were devised as generalizations of finite state automata but with transitions whose selection was governed by a probability distribution. Later Rabin introduced randomized algorithms, and notably the Miller-Rabin Primality Test that up to small probability of error of delivering a false positive, can detect whether a number is prime. These automata can be modeled as probabilistic Turing machines, with varying notions of simulation - Monte Carlo, Las Vegas, etc. - that provide a range of models of probabilistic computation. Randomized automata is now a well-developed field with a number of generalizations and a broad range of applications that include crypto-protocol analysis and other areas of security, among many others. More recently, research in probabilistic programming is an emerging area [3] with applications in Artificial Intelligence, and Machine Learning in particular. Notably, Prakash’s work also has applications here, in the form of Markov Decision Processes. Finally, probability theory is fundamental to quantum computation and to modeling quantum programming languages.

In this month's column, Prakash describes the early history of probabilistic bisimulation leading up to its logical characterization. This notion was originally due to Larsen and Skou in their seminal paper on probabilistic testing [2], but the characterization Prakash describes comes from his work with a number of colleagues that significantly simplified the original logic.

As Prakash points out, the first use of probability in semantic models is due to Dexter Kozen, who introduced a semantic model for probabilistic programs and then developed probabilistic dynamic logic in the early to mid-1980s. Prakash notes that Kozen also introduced measure theory and functional analysis, the traditional mathematical approaches to probability theory, into computer science, an approach that underlies much of his work.

Given the prominent role of the lambda calculus and its models in the development of semantics, one might expect to see a long line of work on probabilistic lambda calculus and associated semantic models. Unfortunately, this is not the case – only recently
has work begun to appear on this theme [1; 6] (but one also notes the earlier work of Ramsey and Pfeffer [5]). One impediment has been the lack of a convincing model for probabilistic computation in domain theory. Indeed, except for the most general category of directed complete partial orders, there appear to be no Cartesian closed categories that support a probability monad that leaves the category invariant.

REFERENCES
This is a very personal review of the definition of probabilistic bisimulation on continuous state spaces and the logical characterization theorem. I only consider fully probabilistic reactive systems.

1. INTRODUCTION
I am delighted to have the chance to write this column at Mike Mislove’s invitation. I have worked on probabilistic systems for about 20 years now and it is interesting to look back and see how far the subject has developed. It is hopeless in the context of a column like this to survey everything that has happened. I have chosen to write about the parts of the subjects in which I was personally involved: this is not to belittle the other contributions to the subject. I hope the people responsible for those other developments will be inspired to write their own columns eventually.

Before I begin, let me mention the many people who have helped me understand this subject. Dexter Kozen was my first teacher in this area and his ideas remain an amazingly fertile source of inspiration. His work is described in two papers from the 1980s where he developed a semantics for probabilistic programs [Kozen 1981] and later a paper where he developed a probabilistic dynamic logic [Kozen 1985]. In the latter he describes a probabilistic analogue of the duality between predicate-transformer semantics and state-transformer semantics. As far as I know, Dexter was the first to introduce ideas from measure theory and functional analysis into computer science papers on this topic.

In my own work I have benefitted immensely from the collaborations with two of my former students: Josée Desharnais and Radha Jagadeesan as well as Vineet Gupta. In the early stages I also collaborated with Rick Blute and Abbas Edalat. Later I got to work with Costin Bădescu, Philippe Chaput, Gheorghe Comanici, Norm Ferns, Kim Larsen, François Laviolette, Radu Mardare, Gordon Plotkin and Doina Precup who introduced me to the world of machine learning. I have to mention Vincent Danos as being particularly influential and finally, at last, I got a chance to collaborate with Dexter (and Radu Mardare and Kim Larsen) on a fun paper on Stone duality for Markov processes. I have also greatly benefitted by conversations and discussions on this and related topics with many others including Samson Abramsky, Luca de Alfaro, Christel Baier, Filippo Bonchi, Frank van Breugel, Ernst-Erich Doberkat, Martin Escardo, Jean Goubault-Larrecq, Holger Hermanns, Bart Jacobs, Achim Jung, Joost-Pieter Katoen, Matteo Mio, Mike Mislove, Daniela Petrisan, Roberto Segala, Alexandra Silva, Ana Sokolova, Ben Worrell, Mingsheng Ying and Chun Lai Zhou.

2. PROBABILISTIC SYSTEMS
The systems I discuss below are inspired by the work of Larsen and Skou [Larsen and Skou 1991] which was a landmark in this area. These systems are labelled transi-
tion systems where the final state is governed by a (sub)probability distribution. Thus, given a current state and an action performed by (or on) the system there is a sub-probability distribution over the possible final states. I allow for the possibility that the action may not terminate or can be rejected so that the sum of the probabilities over the final states may be less than 1; hence the term sub-probability distribution. More importantly, there is no indeterminacy other than that described by the probability. In other words, I am not considering models that combine probability and non-determinism as, for example, in Segala’s models [Segala 1995] or in [Keimel et al. 2009] or [Mislove 2000]. There are many excellent papers on this difficult topic and research on this continues to this day. Furthermore, there is no attempt to assign probabilities to the “external” actions. The probabilities are all conditional on a specific action occurring. This is called a reactive model [Larsen and Skou 1991] in contrast to the generative model [van Glabbeek et al. 1990].

Here is a first cut at the formal definition of the kind of systems I am talking about.

**DEFINITION 1.** A probabilistic labelled transition system is a tuple

\[(S, A, \forall a \in A, T_a : S \rightarrow \text{Sub}(A)),\]

where \(S\) is a finite (or countable, if you really want) set of states, \(A\) is a finite set of actions and \(T_a\) is a transition function and \(\text{Sub}(A)\) denotes the sub-probability distributions on \(A\).

The idea is that \(T_a(s)(s')\) is the probability that the system is in state \(s\) and the action \(a\) is performed the final state will be \(s'\). It is important that these probabilities depend on the action and the current state only; they do not depend on the past history of the system. This is the all important Markov property. It is often convenient to write this function in its uncurried form as \(T_a(s, s')\) in which case it looks like a matrix; indeed it is often called the transition probability matrix. What is the probability that the system starts in \(s_0\) and ends up in \(s_1\) after an \(a\)-transition followed by a \(b\)-transition? Here we do not care where it was immediately after the \(a\)-transition. Clearly this is just

\[\sum_{s \in S} T_a(s_0, s)T_b(s, s_1).\]

In short, one composes the effects of transitions by matrix multiplication.

It is time for some examples. We label the arrows with action names as well as (in parentheses) the numerical probabilities associated with the transition.

```
\[
\begin{array}{c}
s_0 \\
\downarrow a(\frac{1}{4}) \\
\end{array}
\quad \begin{array}{c}
s_1 \\
\downarrow a(\frac{3}{4}) \\
\end{array}
\quad \begin{array}{c}
s_2 \\
\uparrow a(1) \\
\end{array}
\quad \begin{array}{c}
s_3 \\
\end{array}
\end{array}
```

Here, only one action is possible, \(s_1\) is a dead state. If we did not allow sub-probabilities we could not have dead states. The next example has three different action types.
Note that from $s_0$ there are two actions possible and each one has probability 1; each action carries its own probability distribution. This time $s_1$ is not a dead state.

3. PROBABILISTIC BISIMULATION

I assume that the reader is familiar with the concept of bisimulation for states of an ordinary labelled transition system. In the case of probabilistic transition systems we are also trying to define an equivalence relation that captures behavioural indistinguishability. Let us proceed by first considering an example.

Here we have two systems but we can regard them as one and ask, “is it possible to distinguish $s_0$ from $t_0$ by observing the probabilities with which actions are accepted or rejected?” We see that from $t_0$ there is a $\frac{1}{3}$ probability that an $a$-action will lead to a dead state; we observe exactly the same thing from $s_0$. We say that the $t_0 \xrightarrow{a} t_1$ transition is matched by the $s_0 \xrightarrow{a} s_1$ transition in both label and probability. What about the $t_0 \xrightarrow{a} t_2$ transition? There is no transition that matches it from $s_0$. However, we see that the states $s_2$, $s_3$ and $t_2$ are all indistinguishable. So from $s_0$ we cannot tell whether we end up in $s_2$ or $s_3$. If we add the probabilities to take an $a$-transition to these states we get a composite transition that does match the $t_0 \xrightarrow{a} t_2$ transition. Thus, we should first try to find equivalence classes and then match the transition probability to the equivalence classes.

Here is the precise definition due to Larsen and Skou [Larsen and Skou 1991]. We have used a standard overloading of the notation for $T$ by writing $T_a(s, U)$, where $U \subseteq S$, for $\sum_{u \in U} T_a(s, u)$.

**Definition 2.** Given a probabilistic labelled transition system (PLTS) $(S, A, \{T_a\})$, we say that an equivalence relation $R$ on $S$ is a **bisimulation relation** if whenever $sRt$ for states $s, t$ then for every $a \in A$ and every equivalence class $C$ of $R$ we have

$$T_a(s, C) = T_a(t, C).$$

We say that $s$ is **bisimilar to** $t$ if there is some bisimulation relation $R$ such that $sRt$.

---

1Note the indefinite article.
In order to show that two states are bisimilar one has to find a bisimulation relation relating the two states. This may require ingenuity depending on the situation. What if one suspects that the two states are indeed not bisimilar? First we define a modal logic which gives formulas that are defined at the states of a PLTS \((S, A, \{T_a \mid a \in A\})\).

\[
\mathcal{L} ::= T \mid \phi_1 \land \phi_2 \mid \neg \phi \mid (a)^q \phi.
\]

Here \(a\) is an action and \(q\) is a rational number between 0 and 1. The semantics of formulas is as follows:

- \(s \models T\) for every \(s\)
- \(s \models \phi_1 \land \phi_2\) if \(s \models \phi_1\) and \(s \models \phi_2\)
- \(s \models \neg \phi\) if \(s \not\models \phi\)
- \(s \models (a)^q \phi\) if \(T_a(s, \llbracket \phi \rrbracket) \geq q\), where \(\llbracket \phi \rrbracket = \{s \mid s \models \phi\}\).

In words, the last formula says that the state \(s\) can perform an \(a\)-transition and end up in a state satisfying \(\phi\) with probability at least \(q\).

The following theorem due to Larsen and Skou [Larsen and Skou 1991] gives a logical characterization of probabilistic bisimulation.

**Theorem 3.** Let \((S, A, \{T_a \mid a \in A\})\) be a PLTS in which \(S\) is a countable discrete space and there is a finite bound on the branching at every transition and in which every probability is a multiple of a fixed real number \(\varepsilon\). Then \(s\) is bisimilar to \(t\) if and only if

\[
\forall \phi \in \mathcal{L}, s \models \phi \iff t \models \phi.
\]

This is a typical Hennessy-Milner-type [Hennessy and Milner 1980; 1985] theorem due independently to van Benthem [van Benthem 1976]. This probabilistic version requires restrictions on the branching as in the original Hennessy-Milner theorem for nondeterministic processes; in fact the restrictions here are even a bit stronger. Larsen and Skou [Larsen and Skou 1991] established the same theorem for variations of the basic logic with something weaker than full-blown negation. For example, one can drop the negation and introduce a formula \(\Delta_a\), \(s \models \Delta_a\) if the system cannot perform an \(a\)-action in the state \(s\). However, all the logics they present do require some kind of negative formula.

Such a theorem answers the question I raised earlier. In order to tell that two states are not bisimilar it suffices to find a formula on which they disagree.

How important is the relaxation to sub-probabilities? If all the distributions are full probability distributions then every action is enabled with probability 1 in every state. Our notion of observation is that when we attempt an action we can tell whether it is accepted or rejected and, if the latter, with what probability. We cannot see any other aspect of the state. Thus, all we will see is that every action is always accepted. In that case every state is bisimilar to every other and the system collapses to a one-state system!

A proof of this theorem appears in the original paper by Larsen and Skou [Larsen and Skou 1991] and a clever proof, due to James (Ben) Worrell, appears in a book chapter I wrote [Panangaden 2011].

**4. MOTIVATING CONTINUOUS-STATE SPACE SYSTEMS**

The systems that I will describe in the rest of this article - continuous state space and discrete time - are of interest for two reasons. First, this is a reasonable middle ground between the discrete and a completely continuous model. It is clear that continuous
models will force us to make a significant deepening of the mathematics. Second, these systems occur in natural examples. A significant one is the treatment of avionics software. Here the system is coded as a loop with a certain fixed periodicity. In every cycle some actions are taken and some interactions occur. The system itself is inherently continuous but the temporal evolution has a discrete time step imposed by the main control loop. The discrete time, continuous space model is how many engineers conceptualize the system. Of course the physics one is modelling occurs in continuous time, hence an understanding of continuous time would be very useful and important in the future.

The notion of bisimulation is central to the study of concurrent systems. While there is a bewildering variety of different equivalence relations between processes (two-way simulation, trace equivalence, failures equivalence and many more), bisimulation enjoys some fundamental mathematical properties, most notably its characterization as a fixed-point, which make it the most discussed process equivalence. Of course there are many different variants of bisimulation itself! In the present paper we are not so much concerned with adjudicating between the rival claims of all these relations, but rather, we are concerned with showing how to extend these ideas to the world of continuous state spaces. As we shall see below, new mathematical techniques (from the point of view of extant work in process algebra) have to be incorporated to do this. Once the model and the new mathematical ideas have been assimilated, the whole gamut of process equivalences can be studied and discussed.

One might take the view that any automated analysis or logical reasoning must be inherently discrete in character. In particular, even if one is interested in reasoning about a physical system, one has to first discretize the system. In fact, this point of view actually provides a good argument for retaining the continuous view of the system. A given system may well be described in continuous terms. Without formalizing the continuous system, and having a notion of equivalence between discrete and continuous systems how does one argue that the discretized system is a faithful model of the underlying continuous system? Even assuming one is willing to treat a discrete model as given, what if one needs to refine the model? For example, a given discretization may arise from some type of approximation based on a given tolerance; how does one refine the tolerance or discretize adaptively? Clearly the underlying continuous model has to be carried around in a formal way if we are going to construct different discrete approximations.

The analysis of continuous state space systems began with [Blute et al. 1997] and was continued in a series of papers in collaboration with Desharnais, Gupta and Jagadeesan [Desharnais et al. 1998; Desharnais et al. 1999; 2000; Desharnais et al. 2002; Desharnais et al. 2002a; 2002b; 2003; Desharnais and Panangaden 2003; Desharnais et al. 2004] and with later papers with Vincent Danos and others [Danos and Desharnais 2003; Danos et al. 2006; Chaput et al. 2014]. Significant work was done by Franck van Breugel, James Worrell, Dusko Pavlovic and Michael Mislove together with other collaborators [van Breugel and Worrell 2001b; 2001a; Mislove et al. 2004; van Breugel et al. 2005; D. Pavlovic and Worrell 2006]. Another approach based on ultrametric spaces was developed by de Vink and Rutten [de Vink and Rutten 1997; 1999]. This was important for pushing the co-algebraic point of view but ultrametric spaces are not representative of the connected spaces, like manifolds, that arise in physical systems.

5. A SURVEY OF MEASURE THEORY

In a recent lecture at a Winter School in Estonia I was accused by a student of trying to compress a year of analysis into an hour. Here I will attempt to compress a textbook on analysis into two pages! This section can be omitted if the reader is familiar with basics
of measure and probability theory on arbitrary measurable spaces. We assume that the reader knows the basic ideas of measure theory and probability as expounded in, for example “Probability and Measure” by Billingsley [Billingsley 1995] or “Real Analysis and Probability” by Ash [Ash 1972] or the book with the same title by Dudley [Dudley 1989] or “Introduction to Measure and Probability” by Kingman and Taylor [Kingman and Taylor 1966]. My personal favourite is Billingsley’s book.

In measure theory one tries to associate a numerical notion of “size” to sets. This notion is intended to generalize the familiar notion of length of an interval to more intricate sets. The basic inspiration is geometric but the concept makes sense on spaces that have no familiar geometric structure. For probability theory it is clear that such a notion is crucial if one wants to talk about probabilities of events that may be subsets of some continuum. It is important to recognize that the cardinality concept, which is so successful in the discrete case, is completely inappropriate when the state space is a continuum.

In formulating the notion of Markov processes, we need to refine two concepts that were used in the discrete case. We cannot simply define transition probabilities between states; except in rare cases, such transition probabilities are zero. Accordingly, first we have to define transition probabilities between a state and a set of states. Second, we cannot define transition probabilities to any arbitrary set of states; we need to identify a family of sets for which transition probabilities can be sensibly defined. These are the measurable sets.

**DEFINITION 4.** A σ-algebra on a set X is a family of subsets of X which includes X itself and which is closed under complementation and countable unions.

A set equipped with a σ-algebra is called a measurable space. Given a topological space \((X, T)\), we can define the σ-algebra, often written \(B\), generated by the open sets (or, equivalently, by the closed sets). This is usually called the Borel algebra associated with the topology. When there is no confusion about the topology, it is usually just called “the Borel algebra.”

**DEFINITION 5.** Given a σ-algebra \((X, \Sigma)\), a sub-probability measure on X is a \([0, 1]\)-valued set function, \(\mu\), defined on \(\Sigma\) such that

\[- \mu(\emptyset) = 0,\]
\[- \text{for a pairwise disjoint, countable collection of sets,} \{A_i|i \in I\}, \text{in} \Sigma, \text{we require} \]
\[\mu\left(\bigcup_{i \in I} A_i\right) = \sum_{i \in I} \mu(A_i).\]

In addition, for probability measures we require \(\mu(X) = 1\).

I will always be talking about sub-probability measures. Any measure which assigns a finite value to the whole space is called a finite measure. Sub-probability measures are, of course, finite.

**DEFINITION 6.** A function \(f : (X, \Sigma_X) \to (Y, \Sigma_Y)\) between measurable spaces is said to be measurable if \(\forall B \in \Sigma_Y. f^{-1}(B) \in \Sigma_X\).

A key ingredient in the theory is the transition probability function.

**DEFINITION 7.** A transition probability function on a measurable space \((X, \Sigma)\) is a function \(\tau : X \times \Sigma \to [0, 1]\) such that for each fixed \(x \in X\), the set function \(\tau(x, \cdot)\) is a (sub)probability measure, and for each fixed \(A \in \Sigma\) the function \(\tau(\cdot, A)\) is a measurable function.
One interprets $\tau(x, A)$ as the probability of the system starting in state $x$ making a transition into one of the states in $A$. The transition probability is a conditional probability; it gives the probability of the system being in one of the states of the set $A$ after the transition, given that it was in the state $x$ before the transition. In general the transition probabilities could depend on time, in the sense that the transition probability could be different at every step (but still independent of past history); we will only consider the time-independent case.

How do we know that we can construct such functions? It turns out that such functions always exist on certain topological spaces called Polish spaces. These, however, are not quite general enough for our purposes so we will need a broader class of spaces called analytic spaces.

**Definition 8.** A **Polish** space is the topological space underlying a complete, separable metric space; i.e. it has a countable dense subset.

**Definition 9.** An **analytic** space is the image of a Polish space under a continuous function from one Polish space to another.

The following proposition [Dudley 1989] gives equivalent definitions of analytic set.

**Proposition 10.** Suppose that $X$ and $Y$ are Polish spaces and $f$ is a function from $X$ to $Y$. The following are equivalent:

- $f$ is continuous and $A$ is the image of $X$ under $f$,
- $f$ is measurable and $A$ is the image of $X$ under $f$,
- $f$ is continuous and $A$ is the image of a Borel subset $B$ of $X$,
- $f$ is measurable and $A$ is the image of a Borel subset $B$ of $X$,

Thus in this definition it turns out to be equivalent to say “measurable” image and it makes no difference if we take the image of the whole Polish space or of a Borel subset of the Polish space.

Analytic spaces are more general than Polish spaces but they also have the basic property that the transition probability functions that we want can be defined on them.

We say that a $\sigma$-algebra **separates points** if, whenever $x \neq y$, there is a measurable set, $E$, for which $\chi_E(x) \neq \chi_E(y)$. The following powerful theorem and its consequences play a key role in our treatment of the logic. The first theorem is a very strong “rigidity” property: it says that if one has a sub $\sigma$-algebra, say $\Lambda$ of an analytic space $\Sigma$ and $\Lambda$ separates points (so it is not too small) and countably generated (so it is not too large) then $\Lambda$ is all of $\Sigma$.

**Theorem 11.** Let $(X, \Sigma)$ be an analytic space and suppose that $\Sigma_0$ is a countably generated sub $\sigma$-algebra of $\Sigma$ that separates points in $X$. Then $\Sigma_0 = \Sigma$.

It is perhaps a bit misleading to think of countably generated as being a “size” condition. The $\sigma$-algebra consisting of the countable and co-countable subsets of the reals is a sub $\sigma$-algebra distinct from the usual Borel algebra, it is not, of course, countably generated.

Analytic spaces can be quotiented by “nice” equivalence relations.

**Proposition 12.** Let $X$ be an analytic space and let $\sim$ be an equivalence relation on $X$. Assume that there is a sequence $f_1, f_2, \ldots$ of real-valued measurable functions on $X$ such that for all $x, y$ in $X$ we have $x \sim y$ iff for all $f_i$ we have $f_i(x) = f_i(y)$. Then $X/\sim$ is an analytic space.
6. PROBABILISTIC SYSTEMS ON CONTINUOUS STATE SPACES

Recall the definition of an ordinary (non-probabilistic) labelled transition system. We have a state space $S$, a set of actions $A$ and, for each action $a \in A$, we have a transition relation $\xrightarrow{a} \subset S \times S$. For the probabilistic case we need some probabilistic analogue of the notion of relation [Lawvere 1964; Panangaden 1998; 1999]. This is provided by the notion of Markov kernel [Ash 1972] or stochastic kernel [Feller 1968].

**Definition 13.** A Markov kernel $\tau$ on a measurable space $(S, \Sigma)$ is a function of type $S \times \Sigma \rightarrow [0,1]$ such that for each fixed $s \in S$ the function $\tau(s, \cdot) : \Sigma \rightarrow [0,1]$ is a sub-probability measure, and for each fixed measurable set $A \in \Sigma$ the function $\tau(\cdot, A) : S \rightarrow [0,1]$ is measurable.

Why are these the analogues of relations? I will defer the answer to this question to the next section.

Markov kernels are also called “regular conditional probability densities” in the literature. It is not immediately obvious that we can construct such gadgets but we have the following theorem, which partly explains my digression into analytic spaces in the previous section.

**Theorem 14.** Regular conditional probability densities exist on analytic spaces.

This result can be found in the comprehensive textbook of J. Hoffman-Jørgensen [Hoffman-Jørgenson 1994].

When we define labelled Markov processes (LMP), instead of an arbitrary $\sigma$-algebra structure on the set of states, I will require that the set of states be an analytic space and the $\sigma$-algebra be the Borel algebra generated by the topology.

**Definition 15.** A labelled Markov process with label set $A$ is a structure $(S, \Sigma, \{\tau_a \mid a \in A\})$, where $S$ is the set of states, which is assumed to be an analytic space, and $\Sigma$ is the Borel $\sigma$-algebra on $S$, and

$$\forall a \in A, \tau_a : S \times \Sigma \longrightarrow [0,1]$$

is a Markov kernel called the transition sub-probability function.

We will fix the label set to be some $A$ once and for all. We will write $(S, \Sigma, \tau_a)$ for labelled Markov processes, instead of the more precise $(S, \Sigma, \{\tau_a \mid a \in A\})$ and often refer to a process by its set of states.

Here is an example of a continuous-state system from [Desharnais et al. 2002a]. Consider a process with two labels $\{a, b\}$. The state space is upper right quadrant of the real plane, $\mathbb{R}^2$ together with a single extra point. In order to describe this system conveniently we will pretend, at first, that the state space is the entire real plane. When the process makes an $a$-move from state $(x_0, y_0)$, it jumps to $(x, y_0)$, where the probability distribution for $x$ is given by the density $K_a \exp(-\alpha(x - x_0)^2)$, where $K_a = \sqrt{\alpha/\pi}$ is the normalizing factor. When it makes a $b$-move it jumps from state $(x_0, y_0)$ to $(x_0, y)$, where the distribution of $y$ is given by the density function $K_b \exp(-\beta(y - y_0)^2)$.

The meaning of these densities is as follows. The probability of jumping from $(x_0, y_0)$ to a state with $x$-coordinate in the interval $[s, t]$ under an $a$-move is $\int_s^t K_a \exp(-\alpha(x - x_0)^2)dx$. All points with $x < 0$ or $y < 0$ are identified as a single absorbing state. Once it is in this state no more transitions are possible. Note that the probability of jumping to any given point is, of course, 0. In this process the interaction with the environment controls whether the jump is along the $x$-axis or along the $y$-axis but the actual extent of the jump is governed by a probability distribution. If there were just a single label we would have an ordinary (time-independent) Markov process; in fact it would be a Brownian motion with absorbing walls.
7. THE LAWVERE-GIRY MONAD AND PROBABILISTIC RELATIONS

The nicest categorical way to look at LMPs is as coalgebras of a suitable monad. The monad for this coalgebra was defined by Giry [Giry 1981] following suggestions of Lawvere. Lawvere in fact defined the monad in 1964 in an unpublished note [Lawvere 1964] but monads were not invented in 1964 so he could not call it a monad. Giry’s paper says much more than Lawvere’s note, for example she develops the monad also on the category of Polish spaces which requires entirely new and non-trivial proofs.

We start with the category Mes of measurable spaces: \((X, \Sigma_X)\) where \(X\) is a set and \(\Sigma_X\) is a \(\sigma\)-algebra on \(X\), with measurable functions. Recall that a function \(f: (X, \Sigma_X) \to (Y, \Sigma_Y)\) is measurable if \(f^{-1}(B) \in \Sigma_X\) for all \(B \in \Sigma_Y\).

We define the functor \(\Pi: \text{Mes} \to \text{Mes}\) as follows. On objects
\[
\Pi(X) = \{\nu | \nu\text{ is a sub-probability measure on } X\}.
\]
For any \(A \in \Sigma_X\) we get a function \(p_A: \Pi(X) \to [0, 1]\) given by \(p_A(\nu) = \nu(A)\). The \(\sigma\)-field structure on \(\Pi(X)\) is the least \(\sigma\)-field such that all the \(p_A\) maps are measurable. A measurable function \(f: X \to Y\) becomes \(\Pi(f)(\nu) = \nu \circ f^{-1}\). Clearly \(\Pi\) is a functor.

We claim that \(\Pi\) is a monad. We define the appropriate natural transformations \(\eta: I \to \Pi\) and \(\mu: \Pi^2 \to \Pi^2\) as follows:
\[
\eta_X(x) = \delta(x, \cdot), \mu_X(\Omega) = \lambda B \in \Sigma_X. \int_{\Pi(X)} p_B \Omega.
\]
The definition of \(\eta\) should be clear but the definition of \(\mu\) needs to be explained. First, note that \(\Omega\) is a measure on \(\Pi(X)\). Recall that \(p_B\) is the measurable function, defined on \(\Pi(X)\), which maps a measure \(\nu\) to \(\nu(B)\). The \(\sigma\)-field on \(\Pi(X)\) has been defined precisely to make this a measurable function. Now the integral \(\int_{\Pi(X)} p_B \Omega\) is meaningful. Of course one has to verify that \(\mu_X(\Omega)\) is a sub-probability measure. The only slight subtlety is checking that countable additivity holds.

**Theorem 16.** [Giry] The triple \((\Pi, \eta, \mu)\) is a monad on Mes.

The Kleisli category of this monad is called SRel, for stochastic relations. Just as the category of binary relations is the Kleisli category of the powerset monad so the stochastic relations are the Kleisli morphisms for Giry’s “fuzzy” powerset monad. In Giry’s paper she worked with the set of probability distributions rather than with sub-probability distributions as I am doing here. This gives SRel some additional structure (partially additive structure) but it is not important for our purposes [Panangaden 2009].

Here is the explicit description of SRel. The objects are the same as in Mes. An SRel morphism \(h: X \to Y\) is a measurable function from \(X\) to \(\Pi(Y)\). In other words \(h: X \to (\Sigma_Y \to [0, 1])\), applying some Curry powder we get \(h: X \times \Sigma_Y \to [0, 1]\). Now \(h(x, \cdot)\) is clearly a measure on \(Y\) for each fixed \(x\) and, for each fixed \(B \in \Sigma_Y\), \(h(\cdot, B)\) is a measurable function. We see that an LMP is just a Kleisli morphism from \(X\) to itself. In other words, an LMP is a coalgebra of the functor \(\Pi\). Composition is given as follows. Suppose that \(h\) is as above and \(k: (Y, \Sigma_Y) \to (Z, \Sigma_Z)\). Then we define \(k \circ h: (X, \Sigma_X) \to (Z, \Sigma_Z)\) by the formula \((k \circ h)(x, C) = \int_Y k(y, C) h(x, dy)\).

Note how, in this formula, the measurability conditions on \(h\) and \(k\) are just right for the integral to make sense. Note also how this composition formula is a quantitative analogue of relational composition with integration playing the role of the usual existential quantification. One can also see that the composition formula is the uncountable analogue of matrix multiplication.

\[\text{Try not to confuse } \mu \text{ with a measure.}\]
8. THE LOGICAL CHARACTERIZATION OF PROBABILISTIC BISIMULATION

The final topic of this column is the logical characterization of bisimulation for the case of continuous state spaces systems. First we need to define bisimulation for this case. It turns out that the Larsen-Skou definition needs to be modified only slightly. In our first paper [Blute et al. 1997; Desharnais et al. 1998; 2002] we used the notion of “spans of zigzags” inspired by the way bisimulation is treated in modal logic, see for example [Popkorn 1994]. However, later [Desharnais et al. 2003] we realized that the following definition does the trick.

**Definition 17.** Let $S = (S, \Sigma, A, \{\tau_a\})$ be a labelled Markov process. An equivalence relation $R$ on $S$ is a bisimulation if whenever $sRs'$, with $s, s' \in S$, we have that for all $a \in A$ and every $R$-closed measurable set $X \in \Sigma$, $\tau_a(s, X) = \tau_a(s', X)$. Two states are bisimilar if they are related by a bisimulation relation.

Note that the only measure theory consideration is that we restrict attention to $R$-closed measurable sets\(^3\).

Now LMPs on continuous state spaces cannot be finitely branching or rarely even countably branching. One may prima facie expect that there is no hope of a logical characterization result; logic is just too “discrete” for this situation. It was a surprise that we were able to prove a logical characterization theorem but even more shocking one needs no restriction on the branching at all, and perhaps most surprisingly, one can prove the logical characterization result with the following logic:

$$L_0 ::= T \mid \phi_1 \land \phi_2 \mid \langle a \rangle \phi.$$

What is striking is there is no negation nor indeed any negative construct at all! Thus, since the discrete systems are a special case of the continuous ones, one obtains a significant strengthening of the discrete result along the way. The theorem is

**Theorem 18.** Let $(S, \Sigma, A, \{\tau_a\})$ be an LMP with $S$ an analytic space and $\Sigma$ its Borel $\sigma$-algebra. Two states $s, s'$ are bisimilar if and only if for every formula $\phi$ of the logic $L_0$ we have

$$s \models \phi \iff s' \models \phi.$$

Note that this result requires that the underlying state space be an analytic space. We have traded the restrictions on finite branching for the restriction to analytic spaces. How serious is this? Well every manifold is a Polish space and hence certainly analytic. One can construct examples of non-analytic spaces but one would be hard pressed to give an example that is relevant to any system that one would consider in practice.

It is not my intention to prove this theorem here, my book [Panangaden 2009] gives the proof in detail. The proof really exploits the remarkable properties of analytic spaces. The basic idea is to define logical equivalence $\approx$ of states in the evident way. Then one has to show that this relation is in fact a bisimulation. The proof has some twists and turns but it hinges on the fact that quotients of analytic spaces by good equivalence relations (whatever that might mean) are also analytic and also hinges on Theorem 11. Roughly speaking, one can show that if two states $s, t$ are logically equivalent then transition probabilities are the same to sets of states that are of the form $[\phi]$, i.e. that for any action $a$ we have $\tau_a(s, [\phi]) = \tau_a(t, [\phi])$. Now one can leverage the unique structure theorem (Theorem 11) to show that they are the same to all $\approx$-closed.

---

\(^3\)A set $U$ is $R$-closed if $x \in U$ and $xRy$ implies that $y \in U$. Since $R$ is an equivalence relation this means it is a union of equivalence classes.
measurable sets. A very thorough and deep treatment of the subject (far deeper than in my book) is available in an excellent book by Doberkat [Doberkat 2007].

As partial compensation for not showing the proof of the logical characterization theorem let me instead debunk a putative counter-example. Let us step back to the world of nondeterministic systems with the ordinary (non-probabilistic) Hennessy-Milner logic. Here are two states $s_0, t_0$ that cannot be distinguished without negation; it is easy to see this by induction on the structure of positive formulas.

$$s_0 \xrightarrow{a} s_0 \xrightarrow{a} s_1 \xrightarrow{b} s_3 \xrightarrow{b} t_0$$

The formula that distinguishes them is $\langle a \rangle (\neg b) T$.

Why isn’t this the basis of a counter-example showing that the claimed theorem is not correct? We add probabilities to the transitions to obtain the situation shown below.

$$s_0 \xrightarrow{a[r]} s_0 \xrightarrow{a[q]} s_0 \xrightarrow{a[r]} s_1 \xrightarrow{b} s_3 \xrightarrow{b} t_0$$

If $p + q < r$ or $p + q > r$ we can easily distinguish $s_0$ and $t_0$: we choose a rational number $x$ such that $p + q < x < r$ (or the other way around if that is the case). Then $t_0 \models \langle a \rangle x T$ but $s_0 \not\models \langle a \rangle x T$. If $p + q = r$ and $p > 0$, which must be the case otherwise $s_0$ and $t_0$ really are bisimilar, then $q < r$ so again choosing $x$ rational between $q$ and $r$ allows us to write the formula $\langle a \rangle \langle b \rangle x T$ which distinguishes them. Of course, this does not replace a proof, it just shows that the intuition from non-determinism does not apply straightforwardly.

9. CONCLUSIONS

I hope I have given some feeling for the nature of probabilistic bisimulation. What are the consequences of the logical characterization result of the last section? First, since the logic is completely negation-free there is a possibility that one can get a logical characterization of simulation. Josée Desharnais gave an example showing that in fact $L_0$ does not characterize simulation. In [Desharnais et al. 2003] it was shown that if one adds disjunction to the logic then one does get a logical characterization of simulation. This proof uses the technology of domain theory; I know of no simple proof.

Why did the proof of the logical characterization have to involve so much topology? It is my belief that the concept of bisimulation is the wrong one! One should have defined it in terms of co-spans rather than of spans. In a paper with Danos, Desharnais and Laviolette [Danos et al. 2006] we defined a “dual” notion that we called event bisimulation and showed that one gets an easy proof of logical characterization for general measure spaces. Then our earlier proof can be regarded as a proof that for analytic spaces event bisimulation and...
ordinary bisimulation coincide. For people working with the category of sets, the difference between bisimulation and event-bisimulation is not really noticeable. These ideas were also explored independently by Bartels et al. [Bartels et al. 2004]. In the other direction, if one insists on the traditional definition of bisimulation then analytic spaces represent a strict limit of the theory according to recent results of Sánchez Terraf [Terraf 2011].

REFERENCES


TABLE OF CONTENTS

* NEWS
   LICS’15 Call for Participation
   ACM SIGLOG Announcement

* DEADLINES
   Forthcoming Deadlines

* CALLS
   WSFM-BEAT 2015 - Call for Papers
   SR 2015 - Call for Papers
   SOAMED - Call for Participation
   LNMR 2015 - Call for Papers
   FSTTCS 2015 - Call for Papers
   BLC 2015 - Call for Papers
   QBF 2015 - Call for Papers
   WoLLIC 2015 - Call for Participation
   CADE 2015 - Call for Participation
   DCM 2015 - Call for Papers
   GBMSC - Call for Participation
   HIGHLIGHTS 2015 - Call for Participation
   LPNMR 2015 - Call for Participation
   ABZ 2016 - Call for Papers, Answers to the case study, Workshops, Tutorials
   COMPLEXITY 2016 - Call for Participation

* JOB ANNOUNCEMENTS
   PhD Studentship in Algorithms and Complexity at Royal Holloway, University of London
   New Doctoral Program on Logical Methods in Computer Science (LogiCS), Vienna

ICALP/LICS’15
Call for Participation
July 4-10, 2015
Kyoto, Japan
http://www.kurims.kyoto-u.ac.jp/icalp-lics2015
EARLY REGISTRATION DEADLINE: June 12 (local time)

* Joint Conference of The 42nd International Colloquium on Automata, Languages, and Programming (ICALP 2015) and The Thirtieth Annual ACM/IEEE Symposium on LOGIC IN COMPUTER SCIENCE (LICS) will be held in July 6-10, 2015, in Kyoto.
* Eleven invited talks and about 240 papers on all areas of theoretical computer science and logic in computer science will be presented at Grand Prince Hotel Kyoto. Also, seven workshops will take place at Kyoto University on July 4 and 5.

* INVITED SPEAKERS
  Piotr Indyk (MIT, USA)
  Ken-ichi Kawarabayashi (NII, Japan)
  Daniel Kifer (Pennsylvania State University, USA)
  Valerie King (University of Victoria, Canada)
  Thomas Moscibroda (Microsoft Research and Tsinghua University, China)
  Anca Muscholl (Universite Bordeaux, France)
  Peter O’Hearn (Facebook and University College London, UK)
  Luke Ong (University of Oxford, UK)
  Andrew Pitts (University of Cambridge, UK)
  Geoffrey Smith (Florida International University, USA)
  Ryuhei Uehara (JAIST, Japan)

* WORKSHOPS
  Workshop on Logic and Computational Complexity (LCC 2015)
  Workshop on Natural Language and Computer Science 3 (NLCS 3)
  Workshop on Quantum Computational Complexity (QCC)
  Workshop on Realistic Models for Algorithms in Wireless Networks (WRAWN)
  Workshop on Syntax and Semantics for Low-Level Languages (LOLA 2015)
  Workshop on the Verification of Higher-Order Programs (HOPA)
  Young Researchers Forum on Automata, Languages and Programming (YR-ICALP 2015)

* REGISTRATION
  The registration page and the hotel reservation page are already open. The early registration deadline is June 12 (local time). For more details please visit our conference webpage: http://www.kurims.kyoto-u.ac.jp/icalp-lics2015

* LOCAL ORGANIZERS
  Masahito Hasegawa (Kyoto University)
  Atsushi Igarashi (Kyoto University)
  Kazuo Iwama (Kyoto University)
  Kazuhisa Makino (Kyoto University)

* CONTACT
  icalp-lics2015@kurims.kyoto-u.ac.jp

ACM SIGLOG ANNOUNCEMENT
http://siglog.acm.org
* The ACM has recently chartered a Special Interest Group on Logic and Computation (ACM SIGLOG). Its first Chair is Prakash Panangaden, the other officers are Luke Ong (vice-Chair), Natarajan Shankar (Treasurer) and Alexandra Silva (Secretary).
* The ACM-IEEE Symposium on Logic in Computer Science 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).

DATES
* ICALP/LICS 2015
  Conference: July 4-10, 2015
  http://www.kurims.kyoto-u.ac.jp/icalp-lics2015/
* WSFM-BEAT 2015
  Abstract submission: July 1, 2015 (Wednesday)
  Paper submission: July 3, 2015 (Friday)
* SR 2015
  Abstract submission: July 1, 2015
  Paper submission: July 3, 2015
  https://sites.google.com/site/sr2015homepage/
* SOAMED
  Graduate School: October 4-9, 2015
  Registration Deadline: July 1, 2015
  https://www.informatik.hu-berlin.de/de/forschung/gbiete/soamed
* LNMR 2015
  Paper registration: July 6, 2015 (EXTENDED)
  Submission deadline: July 10, 2015 (EXTENDED)
  http://lnmr2015.insight-centre.org/
* FSTTCS 2015
  Paper submission: July 13, 2015
  http://www.fsttcs.org
* BLC 2015
  Abstract deadline: July 15, 2015
  http://www.newton.ac.uk/event/blc-2015
* QBF 2015
  Extended abstracts submission: Jul 20 2015
  http://fmv.jku.at/qbf15/
* WoLLIC 2015
  Workshop: July 20-23, 2015
  http://www.indiana.edu/~iulg/wollic/
* CADE 2015
  Call for Participation
  Freie Universitat Berlin, Germany, 1-7 August 2015
  http://www.cade-25.info
* DCM 2015
  Submission Deadline for Extended Abstracts (5 pages): August 3, 2015
  http://dcm-workshop.org.uk/2015/
* GBMSC
  Conference: August 17-28, 2015
  http://booleanconferences.ucc.ie/gbmsc2015
* HIGHLIGHTS 2015
  Conference: Prague, 15-18 September 2015
  http://highlights-conference.org
* LPNMR 2015
  Conference: Lexington, KY, USA, September 27-30, 2015
  http://lpnmr2015.mat.unical.it/
* ABZ 2016
  Workshop proposal submissions: October 16, 2015
  Workshop proposal notifications: November 6, 2015
  Research paper and answers to case study submission: January 15, 2016
  Short paper submission: February 4, 2016
  Tutorial proposal submissions: February 15, 2016
  http://www.cdcc.faw.jku.at/ABZ2016/

* SPECIAL SEMESTER ON COMPUTATIONAL COMPLEXITY
  AND PROOF COMPLEXITY 2016
  April-June 2016
  Chebyshev Laboratory at St. Petersburg State University
  Organized jointly with the Skolkovo Institute of Science and Technology.

WSFM-BEAT 2015 - INTERNATIONAL SYMPOSIUM ON WEB SERVICES, FORMAL METHODS AND BEHAVIOURAL TYPES
  Call for Papers
  September 4-5, 2015, Madrid, Spain

* The Symposium on International Symposium on Web Services, Formal Methods and Behavioural Types (WS-FM/BEAT 2015) results from joining the Workshop on Web Services and Formal Methods (WS-FM) and the Workshop on Behavioural Types. The former was mainly devoted to formal aspects of service-oriented and cloud computing. The latter addressed type languages and systems to specify, characterise, and reason about dynamic aspects of program execution.

Topics of either theoretical or applied interest include, but are not limited to: Foundational aspects of large behavioural software systems; Specification, verification, analysis, and testing of large behavioural software systems; Language-based approaches to large behavioural software systems; Security, trust, QoS, dependability, and privacy in large behavioural software systems; Ontologies, standards and technologies for large behavioural software systems; Case-studies on formal methods in large behavioural software systems; Innovative application scenarios of large behavioural software systems.

* Proceedings: All accepted regular papers will be included in the proceedings, which will be published after the symposium as a volume of the LNCS series. Selected short papers will be invited to the proceedings.

* Important dates:
  - Abstract submission: July 1 2015 (Wednesday)
  - Paper submission: July 3 2015 (Friday)
  - Notification: July 31 2015 (Friday)
  - Camera Ready version: August 14 2015 (Friday)

* Invited Speakers: Cosimo Laneve, University of Bologna, Italy, One more TBA
* Workshop co-chairs: Antonio Ravara, Universidade Nova de Lisboa, Portugal and Jan Martijn van der Werf, Universiteit Utrecht, Netherlands
* Detailed information can be found in the webpage above.

SR 2015 - 3rd INTERNATIONAL WORKSHOP ON STRATEGIC REASONING
  Call for Papers

ACM SIGLOG News  88  July 2015, Vol. 2, No. 3
September 21-22, 2015, Oxford, England
https://sites.google.com/site/sr2015homepage/
* The SR international workshop is an annual event that aims to bring together researchers working on different aspects of strategic reasoning and game-based approaches to logic, semantics, and formal verification in computer science, artificial intelligence, and multi-agent systems, both from a theoretical and a practical point of view.
* Proceedings: We plan to produce a special issue of the Journal of Information and Computation with extended versions of selected papers.
* Important dates:
  - July 1 2015: Abstract submission deadline
  - July 3 2015: Paper submission deadline
  - July 31 2015: Acceptance notification
  - Sept 21-22 2015: SR 2015 Workshop
* Invited Speakers: Johan van Benthem (Stanford/Amsterdam Universities), Joseph Halpern (Cornell University), Marta Kwiatkowska (Oxford University), Moshe Vardi (Rice University)
* Detailed information can be found in the webpage above.

SOAMED 2015
Call for Participation
October 4-9, 2015, Steinhofel Castle (near Berlin), Germany
https://www.informatik.hu-berlin.de/de/forschung/gebiete/soamed
* REGISTRATION:
https://www.informatik.hu-berlin.de/de/forschung/gebiete/soamed/Herbstschule/HS%20SOAMED
* AIMS: The DFG-funded graduate school SOAMED invites applications to participate in its autumn school on "processes and data". This school addresses master students and young PhD students in informatics or related fields, interested in up-to-date research on the integration of service oriented architectures, data management, and their application to healthcare systems. It will be organized in the form of talks and tutorials given by a number of international experts in these fields. The course language is English. Participation is free of charge (travelling not included), but there is only limited space available.
* Organisers: Wolfgang Reisig and Ulf Leser

2ND INTERNATIONAL WORKSHOP ON LEARNING AND NONMONOTONIC REASONING (LNMR 2015)
Call for Papers
27-30 September 2015, Lexington, KY, USA
http://lnmr2015.insight-centre.org/
co-located with the
13th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2015)
http://lpnmr2015.mat.unical.it/
* AIMS AND SCOPE
This workshop follows from its first edition in 2013 in an attempt to provide an open forum for the identification of problems and discussion of possible collaborations among researchers with complementary expertise. To facilitate interactions between researchers in the areas of (machine) learning and nonmonotonic reasoning, we welcome contributions focusing on
problems and perspectives concerning both learning and nonmonotonic reasoning.

* SUBMISSIONS
We solicit original papers which are not published elsewhere. Papers should be written in English and be formatted according to the Springer Verlag LNCS style, which can be obtained from http://www.springeronline.com. Every paper should not exceed 12 pages including the title page, references and figures. All submissions will be peer-reviewed and all accepted papers must be presented at the workshop. Paper submission will be electronic through the LNMR-15 Easychair site: https://easychair.org/conferences/?conf=lnmr2015.

* PROCEEDINGS
Workshop organizers are considering to publish an on-line proceedings in a formal way. The details will be announced later. Based on the quality of submissions, a special journal issue will also be considered.

* IMPORTANT DATES
Paper registration: July 6 (EXTENDED)
Submission deadline: July 10 (EXTENDED)
Notification: August 17
Final version due: September 1
Workshop: 1 or 2 days in September 27-30

* WORKSHOP CO-CHAIRS
Alessandra Mileo, INSIGHT Centre for Data Analytics, NUI Galway, Ireland
Alessandra Russo, Dept. of Computing, Imperial College London, UK

35TH FOUNDATIONS OF SOFTWARE TECHNOLOGY AND THEORETICAL COMPUTER SCIENCE (FSTTCS 2015)
Call for Papers
December 16-18, 2015, Bangalore, India
http://www.fsttcs.org

* AIMS
The FSTTCS conference, organized by IARCS (the Indian Association for Research in Computing Science), is a forum for presenting original results in foundational aspects of Computer Science and Software Technology.

* SCOPE
Representative areas include, but are not limited to, the following: algorithms and data structures, algorithmic graph theory and combinatorics, approximation algorithms, automata and formal languages, combinatorial optimization, communication complexity, computational biology, computational complexity, computational geometry, computational learning theory, cryptography and security, game theory and mechanism design, logic in computer science, model theory, modal and temporal logics, models of concurrent and distributed systems, models of timed, reactive, hybrid and stochastic systems, parallel, distributed, and online algorithms, parameterized complexity, principles and semantics of programming languages, program analysis and transformation, proof complexity, quantum computing, randomness in computing, specification, verification, and synthesis, theorem proving, decision procedures, and model checking, theoretical aspects of mobile and high performance computing.

* IMPORTANT DATES
BRITISH LOGIC COLLOQUIUM (BLC 2015)

Call for Abstracts
Cambridge, England
2-4 September 2015
http://www.newton.ac.uk/event/blc-2015

* MEETING
The 2015 meeting of the British Logic Colloquium will be held in Cambridge on 2nd-4th September. It will be preceded by BLC PhD day (1st-2nd September). This is a general Logic meeting covering a variety of topics within mathematical, philosophical and computer science logic. The meeting will include ten invited talks (speakers listed below) and a number of contributed talks. Anyone wishing to contribute a talk should send an abstract (of about 250 words) to blc-2015@cl.cam.ac.uk by 15 July, 2015.

* REGISTRATION
Registration is now open: http://www.newton.ac.uk/event/blc-2015

* INVITED SPEAKERS
Andreas Blass (Michigan); Victoria Gitman (New York); Ian Pratt-Hartman (Manchester); Alexander Kechris (Pasadena); Jonathan Kirby (East Anglia); Agi Kurucz (London); Itay Neeman (Los Angeles); Arno Pauly (Cambridge); Andrew Pitts (Cambridge); Mehrnoosh Sadrzadeh (London)

* BURSARIES
There is a limited number of bursaries available for students who wish to attend. See webpage for details

* PHD DAY
The BLC PhD day provides an opportunity for postgraduates to meet and discuss their research or area of interest with fellow young logicians. If you wish to attend the PhD day, please email blc-2015-phd@maths.cam.ac.uk, including your Name, Affiliation and whether or not you would like to present a talk or a poster.

QBF 2015 - 3rd INTERNATIONAL WORKSHOP ON QUANTIFIED BOOLEAN FORMULAS

Call for Papers
September 23 2015, Austin, Texas, USA.
http://fmv.jku.at/qbf15/

* The goal of the Third International Workshop on Quantified Boolean Formulas (QBF 2015) is to bring together researchers working on theoretical and practical aspects of QBF solving and applications. The workshop is affiliated to and co-located with the SAT conference 2015, September 24-27, 2015.

* Important dates:
- Jul 20 2015: submission of extended abstracts (max. four pages)
- Aug 12 2015: notification of acceptance
- Sep 23 2015: workshop

* Topics of interest (not limited to the following): QBF applications; encodings; benchmarks; certificates; proofs; proof formats; proof checkers; decision procedures; QBF solving; implementation details; structural QBF solving; heuristics; preprocessing;

* Please see the workshop website for further information:
http://fmv.jku.at/qbf15/
22ND WORKSHOP ON LOGIC, LANGUAGE, INFORMATION AND COMPUTATION (WoLLIC 2015)
Call for Participation
July 20th-23rd, 2015, Bloomington, IN, USA
http://www.indiana.edu/~iulg/wollic

* AIMS
WoLLIC is an annual international forum on inter-disciplinary research involving formal logic, computing and programming theory, and natural language and reasoning. Each meeting includes invited talks and tutorials as well as contributed papers.

* INVITED SPEAKERS
Adriana Compagnoni (Stevens Institute, USA)
Nina Gierasimczuk (University of Amsterdam)
John Harrison (Intel, USA)
Peter Jipsen (Chapman U, USA)
Andre Joyal (U du Quebec Montreal, Canada)
Chung-chieh Shan (Indiana U, USA)
Alexandra Silva (Radboud U Nijmegen, The Netherlands)
Mehrnoosh Sadrzadeh (Queen Mary, UK)

* STUDENT GRANTS
ASL sponsorship of WoLLIC 2015 will permit ASL student members to apply for a modest travel grant (deadline: May 1st, 2015).

* DETAILS
Detailed information can be found on the webpage www.indiana.edu/~iulg/wollic

25TH INTERNATIONAL CONFERENCE ON AUTOMATED DEDUCTION (CADE-25)
Call for Participation
Freie Universitat Berlin, Germany, 1-7 August 2015
http://www.cade-25.info

* CADE is the major forum for the presentation of research in all aspects of automated deduction.

* IMPORTANT DATES:
Early Registration Deadline: 21 June 2015

* INVITED SPEAKERS:
CADE-25 jubilee session
* Ursula Martin, University of Oxford
* Frank Pfenning, Carnegie Mellon University
* David Plaisted, University of North Carolina at Chapel Hill
* Andrei Voronkov, University of Manchester
CADE-25 main conference
* Ulrich Furbach, University of Koblenz, ECCAI invited talk
* Edward Zalta, Stanford University
* Michael Genesereth, Stanford University (joint with RuleML Symposium)
Reception and dinner speeches will be given by Wolfgang Bibel and Jorg Siekmann.

* WORKSHOPS
Bridging: Bridging the gap between human and automated reasoning
DT: 29. Jahrestreffen der GI-Fachgruppe Deduktionssysteme
HOL4: HOL4 Workshop
IWC: The 4th International Workshop on Confluence
LFMTP: International Workshop on Logical Frameworks and
Meta-Languages: Theory and Practice
LOCAS: Low-level Code Analysis for Security
PxTP: Workshop on Proof eXchange for Theorem Proving
QUANTIFY: 2nd International Workshop on Quantification
Vampire: The Vampire Workshop
* TUTORIALS
Abella: Reasoning about Computational Systems using Abella
Beluga: Programming proofs about formal systems
CProVER: From Programs to Logic: The CProVER verification tools
Isabelle: Isabelle Tutorial
Lean: Lean Theorem Prover: a Tutorial
Superposition: 25th Anniversary of Superposition: Status and Future
* COMPETITIONS
CASC: The CADE ATP System Competition
CoCo: The 4th Confluence Competition
termCOMP: Termination Competition
* POSTER EVENT
EPS: The CADE-25 Taskforce towards an Encyclopedia of Proof Systems
* ORGANIZERS:
Conference Chair:
Christoph Benzmueller (Freie Universitat Berlin)
Program Committee Co-Chairs:
Amy Felty (University of Ottawa)
Aart Middeldorp (University of Innsbruck)
Workshop, Tutorial, and Competition Co-Chairs:
Jasmin Blanchette (Technische Universitat Muenchen)
Andrew Reynolds (EPFL Lausanne)
Publicity and Web Chair:
Julian Roder (Freie Universitat Berlin)

11TH WORKSHOP ON DEVELOPMENTS IN COMPUTATIONAL MODELS (DCM 2015)
Call for Papers
October 28, 2015, Cali - Colombia
http://dcm-workshop.org.uk/2015/
Satellite event of ICTAC 2015 (http://www.ictac2015.co)
* AIM
The DCM workshop aims to bring together researchers who are currently
developing new computational models or new features for traditional
computational models, in order to foster their interaction, to provide
a forum for presenting new ideas and work in progress, and to enable
newcomers to learn about current activities in this area.
* TOPICS OF INTEREST
Topics of interest include all abstract models of computation and
their applications to the development of programming languages and
systems. This includes (but is not limited to):
- Functional calculi: lambda-calculus, rho-calculus, term and graph rewriting;
- Quantum computation, including implementations and formal methods in
quantum protocols;
- Probabilistic computation and verification in modelling situations;
Chemical, biological and bio-inspired computation, including spatial models, self-assembly, growth models;
Models of concurrency, including the treatment of mobility, trust, and security;
Infinitary models of computation;
Information-theoretic ideas in computing.
* IMPORTANT DATES
Notification: 13 Sep 2015;
Pre-proceedings version due: 5 Oct 2015
Workshop: 28 Oct 2015
Deadline for EPTCS Proceedings: 7 Dec 2015
* CHAIRS
Cesar Munoz, NASA (USA)
Jorge A. Perez, University of Groningen (NL)

GEORGE BOOLE MATHEMATICAL SCIENCES CONFERENCE (GBMSC 2015)
Call for Participation
17-25 August 2015
Cork, Ireland
http://booleconferences.ucc.ie/gbmsc2015
* CONTEXT
As part of the celebrations of Boole’s bicentenary, the George Boole Mathematical Sciences (GBMS) Conference (including Domains XII) will be held in University College Cork (UCC) during the last two weeks of August 2015. George Boole (1815-1864) was the first professor of mathematics at Cork. Boole’s efforts to mathematize logical thinking caused a lasting paradigm shift in the 19th century which enlarged the scope and potency of modern mathematics, and provided a wealth of ideas for applications in diverse scientific areas resulting in ground-breaking innovations during the 20th century and beyond. This event will include 100-150 lectures on selected areas:
Theme 1: Boole and Beyond in Quantum Information Theory
Theme 2: From Boole’s Algebra of Logic to Boolean Algebra, and Beyond
Theme 3: Complex and Boolean Networks Geometry and Visualization
Theme 4: Geometry and Visualization
Theme 5: Harmonic Analysis
Theme 6: Invariants from Moduli Spaces
Theme 7: Mathematical Financial Modelling Post-Crisis
Theme 8: Quantum Probabilistic Symmetries and Quantized Boolean Algebras
The following events are embedded in and associated with the conference:
2015 Annual Meeting of the Irish Mathematical Society (IMS)
Domains XII
When Boole Meets Shannon

HIGHLIGHTS OF LOGIC, GAMES AND AUTOMATA (HIGHLIGHTS 2015)
Call for Presentations
Prague, 15-18 September 2015
http://highlights-conference.org
* AIMS
HIGHLIGHTS 2015 is the third conference on Highlights of Logic, Games
and Automata which aims at integrating the community working in these fields. A visit to Highlights conference should offer a wide picture of the latest research in the area and a chance to meet everybody in the field, not just those who happen to publish in one particular proceedings volume. The participants present their best work, be it published elsewhere or yet unpublished.

* **DATES**

The conference is three days long (Sept. 16-18) and it is preceded by the Highlights tutorial day (Sept. 15). The contributed talks are around ten minutes. The participation costs are modest (around 80 Euro) and some cheap accommodation close to conference site is arranged. Prague is easy to reach.

* **SUBMISSION DEADLINE**

June 12, 2015


---

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

**Call for Participation**

Lexington, KY, USA, September 27-30, 2015

http://lpnmr2015.mat.unical.it/

* **REGISTRATION:** Registration procedure is available via http://www.cs.uky.edu/lpnmr2015/.

* **AIMS:**

LPNMR 2015 is the thirteenth in the series of international meetings on logic programming and non-monotonic reasoning. LPNMR is a forum for exchanging ideas on declarative logic programming, non-monotonic reasoning, and knowledge representation.

The program will include three invited talks:

- **Stable Models for Temporal Theories** - By Pedro Cabalar, University of Corunna, Spain

- **Algorithmic decision theory meets logic** - By Jerome Lang, Universite Paris-Dauphine, France (Plenary session with ADT 2015).

- **Relational and Semantic Data Mining** - By Nada Lavrac, Jozef Stefan Institute and University of Nova Gorica, Slovenia

This edition of LPNMR will also feature several workshops, a special session dedicated to the 6th ASP Systems Competition, and will be collocated with the 4th Algorithmic Decision Theory Conference, ADT 2015. Joint LPNMR-ADT Doctoral Consortium will be a part of the program.

Full info are available via the official conference website http://lpnmr2015.mat.unical.it/.

* **GENERAL CHAIR:** Victor Marek, University of Kentucky, KY, USA

* **PROGRAM CHAIRS**

Giovambattista Ianni, University of Calabria, Italy
Mirek Truszczynski, University of Kentucky, KY, USA

* **WORKSHOPS CHAIR**

Yuliya Lierler, University of Nebraska at Omaha, NE, USA

---

**5TH INTERNATIONAL ABZ 2014 CONFERENCE (ASM, Alloy, B, TLA, VDM, Z)**
Call for Papers, Answers to the case study, Workshops, Tutorials
May 23-27, 2016
Linz, Austria
http://www.cdcc.faw.jku.at/ABZ2016/

* The ABZ conference is dedicated to the cross-fertilization of six related state-based and machine-based formal methods, Abstract State Machines (ASM), Alloy, B, TLA, VDM and Z. Contributions are solicited on all aspects of the theory and applications of ASMs, Alloy, B, TLA, VDM, Z approaches in software/hardware engineering, including the development of tools and industrial applications.

* Types of submission:
  -- Research papers: full papers of not more than 14 pages in LNCS format, which have to be original, unpublished and not submitted elsewhere.
  -- Short presentations of work in progress, and tool demonstrations. An extended abstract of not more than 4 pages is expected and will be reviewed.
  -- Answers to case study papers: full papers of not more than 14 pages in LNCS format reporting on the experiments conducted with any of the state based techniques in the scope of ABZ 2014.
  -- Application in industry papers reporting on work or experiences on the application of state based formal methods in industry. An extended abstract of not more than 4 pages is expected and will be reviewed.

* Submission site: https://easychair.org/conferences/?conf=abz2016

* Important Dates:
  Workshop proposal submission: October 16, 2015
  Research paper, Answers to case study submission: January 15, 2016
  Short and industry paper submission: February 4, 2016
  Tutorial proposal submissions: February 15, 2016
  Tutorial proposal notifications: March 14, 2016

* Detailed information can be found on the conference website
* Contact: Klaus-Dieter SCHEWE (klaus-dieter.schewe@scch.at)

SPECIAL SEMESTER ON COMPUTATIONAL COMPLEXITY AND PROOF COMPLEXITY
* April-June 2016
* Chebyshev Laboratory at St.Petersburg State University
* Organized jointly with the Skolkovo Institute of Science and Technology.
* Events include a WORKSHOP ON PROOF COMPLEXITY, May 17-20, 2016, St. Petersburg, organized by Sam Buss and Pavel Pudlak, keynote speaker Jan Krajicek; and a WORKSHOP ON LOW-DEPTH COMPLEXITY, May 23-25, 2016, St. Petersburg, organized by Ben Rossman and Rahul Santhanam, keynote speaker Ryan Williams.
* Short courses will be held before each workshop.
* Graduate student, postdocs and other researchers may apply for funding for both short or extended visits throughout the semester.
* To inquire about participation, or apply for funding, please fill out the form on the web page or email the organizers directly.
* Organizers: Sam Buss and Edward A. Hirsch.

PHD STUDENTSHIP IN ALGORITHMS AND COMPLEXITY
AT ROYAL HOLLOWAY, UNIVERSITY OF LONDON

ACM SIGLOG News 96 July 2015, Vol. 2, No. 3
* The Department of Computer Science at Royal Holloway, University of London is offering a three-year full-time PhD studentship in algorithms and complexity starting in October 2015. The studentship includes a full tuition-fee waiver and a maintenance award in line with the level recommended by Research Councils UK (exact value to be confirmed, circa GBP 16,000 (USD 25,000 or EUR 20,000)). The student will be hosted in the Center for Algorithms and Applications and will work under the supervision of Dr Iddo Tzameret (http://www.cs.rhul.ac.uk/home/tzameret/).

* THE PROJECT is broadly in the area of computational complexity with an emphasis on satisfiability and the complexity of proofs. The successful candidate will investigate fundamental aspects of the Boolean satisfiability problem SAT from possibly different aspects - combinatorial, algebraic and logical - with a possibility to engage as well in applied or empirical study of SAT-solving and other applications related to SAT, depending on the preferences and qualifications of the candidate.

* For more information about the post see: http://www.cs.rhul.ac.uk/home/tzameret/PhD.Post.html

* STARTING DATE: October 1, 2015

* HOW TO APPLY:
Applications should be made as soon as possible through the online application system at Royal Holloway, University of London: https://www.royalholloway.ac.uk/studyhere/researchdegrees/applying/home.aspx and will remain open until the position is filled.

* For any informal inquiries about the position, please contact Dr Iddo Tzameret at: Iddo.Tzameret@rhul.ac.uk

NEW DOCTORAL PROGRAM ON LOGICAL METHODS IN COMPUTER SCIENCE (LogiCS)

http://logic.cs.at/phd

* Funded Doctoral Positions in Computer Science

* TU Wien, TU Graz, and JKU Linz are seeking exceptionally talented and motivated students for their joint doctoral program LogiCS. The LogiCS doctoral college focuses on interdisciplinary research topics covering (i) computational logic, and applications of logic to (ii) databases and artificial intelligence as well as to (iii) computer-aided verification.

* THE PROGRAM
LogiCS is a doctoral college focusing on logic and its applications in computer science. Successful applicants will work with and be supervised by leading researchers in the fields of computational logic, databases and knowledge representation, and computer-aided verification.

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

* POSITIONS AND FUNDING
We are looking for 1-2 doctoral students per faculty member, where 30% of the positions are reserved for highly qualified female candidates. The doctoral positions are funded for a period of 3 years according to the funding scheme of the Austrian Science Fund (details: http://www.fwf.ac.at/de/projects/personalkostensaetze.html) The funding can be extended for one additional year contingent on a placement at one of our international partner institutions.

* CURRENT RESEARCH AREAS
At the moment we are particularly looking for people in the following areas:
- Answer Set Programming
- Model Checking
- Proof Theory and Automated Deduction
- QBF-solving
- Static Analysis and Abstract Interpretation

* HOW TO APPLY
Detailed information about the application process is available on the LogiCS web-page http://logic-cs.at/phd/
The applicants are expected to have completed an excellent diploma or master's degree in computer science, mathematics, or a related field. Candidates with comparable achievements will be considered on a case-by-case basis. Applications by the candidates need to be submitted electronically.
Applications can be submitted at any time.
Next application Deadline: July 1, 2015.

* HIGHEST QUALITY OF LIFE
The Austrian cities Vienna, Graz, and Linz, located close to the Alps and surrounded by beautiful nature, provide an exceptionally high quality of life, with a vibrant cultural scene, numerous cultural events, world-famous historical sites, a large international community, a varied cuisine and famous coffee houses.

* For further information please contact: info@logic-cs.at
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: _________________________

Make check or money order payable to ACM, Inc

ACM accepts U.S. dollars or equivalent in foreign currency. Prices include surface delivery charge. Expedited Air Service, which is a partial air freight delivery service, is available outside North America. Contact ACM for more information.

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

www.acm.org/joinsigs

Advancing Computing as a Science & Profession