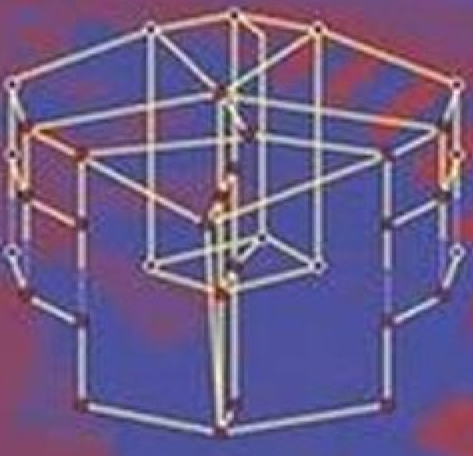
State-of-the-Art Survey

LNCS 5250

Nadia Creignou
Phokion G. Kolaitis
Heribert Vollmer (Eds.)

Complexity of Constraints

An Overview of Current Research Themes



Springer

Lecture Notes in Computer Science 5250

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Nadia Creignou   Phokion G. Kolaitis
Heribert Vollmer (Eds.)

Complexity of Constraints

An Overview of Current Research Themes

Springer

Volume Editors

Nadia Creignou
LIF (CNRS UMR 6166)
Aix-Marseille Université
Marseille, France
E-mail: creignou@lif.univ-mrs.fr

Phokion G. Kolaitis
IBM Almaden Research Center
E-mail: kolaitis@almaden.ibm.com
and
Computer Science Department
University of California, Santa Cruz
E-mail: kolaitis@cs.ucsc.edu

Heribert Vollmer
Institut für Theoretische Informatik
Leibniz Universität Hannover
Hannover, Germany
E-mail: vollmer@thi.uni-hannover.de

Library of Congress Control Number: Applied for

CR Subject Classification (1998): F.2, E.1, G.2, I.2.8, I.3.5, G.1

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-92799-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-92799-0 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12592739 06/3180 5 4 3 2 1 0

Preface

In October 2006, the editors of this volume organized a Dagstuhl Seminar on "Complexity of Constraints" at the Schloss Dagstuhl Leibniz Center for Informatics in Wadern, Germany. This event consisted of both invited and contributed talks by some of the approximately 40 participants, as well as problem sessions and informal discussions. After the conclusion of the seminar, the organizers invited a number of speakers to write surveys presenting the state-of-the-art knowledge in their area of expertise. These contributions were peer-reviewed by experts in the field and revised before they were included in this volume. In addition, this volume contains a reprint of a survey by P.G. Kolaitis and M.Y. Vardi on the logical approach to constraint satisfaction that first appeared in "Finite Model Theory and Its Applications," (Springer 2007).

We thank the Directorate of Schloss Dagtuhl for its support, the speakers of the seminar for making it a successful event, and, above all, the contributors to this volume for their informative and well-written surveys. We also thank Arne Meier for technical assistance during the final compilation of this book, and Alfred Hofmann at Springer for his support and guidance.

July 1 (the birthday of Gottfried Wilhelm Leibniz) 2008

Nadia Creignou
Phokion G. Kolaitis
Heribert Vollmer

Table of Contents

Introduction

The first systematic complexity-theoretic study of constraints was carried out by T.J. Schaefer in 1978 with a paper on Boolean constraint satisfaction problems (CSPs). This volume opens with a survey of Boolean constraints by N. Creignou and H. Vollmer. Schaefer proved a *Dichotomy Theorem* about the satisfiability problem for Boolean CSPs, which asserts that each Boolean CSP is either NP-complete or in P (hence, assuming P $\neq$ NP, all infinitely many intermediate complexity degrees are avoided). Creignou and Vollmer present a modern algebraic proof of Schaefer's Dichotomy Theorem, a proof that makes use of Galois theory and of the structure of the lattice of Boolean clones, known as *Post's lattice*. In addition to the satisfiability problem, several other algorithmic problems, including counting, optimization, and circumscription, can be completely classified from a complexity point of view using Post's lattice. The first contribution to this volume surveys these results and poses the question: what is so special about certain algorithmic tasks that makes Post's lattice applicable to their classification.

The realm of Boolean universes is by now quite well understood. When moving to larger finite domains, the complexity-theoretic study of CSPs relies on Galois connections. The second contribution to this volume, by F. Börner, presents a systematic introduction to Galois theory, to the widely used Pol-Inv Galois connection, and also to variants of this Galois connection that only very recently have turned out to be important for the study of CSPs.

One complication that arises when moving from 2-element universes to larger ones is that the lattice of clones is well-understood only for the former case. As a matter of fact, this lattice becomes uncountable even for 3-element domains. Consequently, even though the algebraic underpinnings remain the same, the situation now becomes much more complicated in a fundamental way. In 1993, T. Feder and M.Y. Vardi articulated their *Dichotomy Conjecture* to the effect that the satisfiability problem will avoid all degrees between 0 (polynomial time) and 1 (NP-complete) in all finite domains. This conjecture was proved by A. Bulatov for 3-element universes in 2002, but it remains open to this day for all domains of cardinality bigger than 3. The survey by A. Bulatov and M. Valeriote describes recent algebraic attacks to the Feder-Vardi Dichotomy Conjecture, leading to some rather unexpected universal-algebraic formulations of this conjecture in the language of tame congruence theory.

The survey by A. Bulatov, A. Krokhin, and B. Larose focuses on *dualities* and their relationship to the Feder-Vardi Dichotomy Conjecture. In fact, the class of CSPs that exhibit the so called bounded treewidth duality is one of the largest classes with a tractable satisfiability problem. Additional large tractable classes can be identified using a logical approach, as described in the survey by Kolaitis and Vardi.

N. Creignou et al. (Eds.): Complexity of Constraints, LNCS 5250, pp. 1–2, 2008.

An important line of research identifies "islands of tractability" for the *uniform constraint satisfaction problem*, where not only the conjunctive query but also the algebraic structure of the constraints is part of the input. Interestingly, this field has a tight connection to database theory, and many algorithmic approaches from there lead to significant progress in the complexity of (uniform) CSPs. This is the topic of the survey by F. Scarcello, G. Gottlob, and S. Greco.

In another variant of CSPs, which has been studied only very recently, one considers the situation where the constraints are defined over an infinite domain. The survey by M. Bodirsky presents important examples of such problems, shows how the universal-algebraic approach finds applications here, and overviews both the main results and certain fundamental questions that remain open.

The contribution by H. Schnoor and I. Schnoor returns to question of determining for which algorithmic problems Post's lattice may help. While the usual Pol-Inv Galois connection is tailored towards the satisfiability problem (and for additional algorithmic tasks), Shnoor and Schnoor show that a similar Galois connection involving clones of partial functions is potentially applicable to a wide range of algorithmic problems. Unfortunately, even in the Boolean case, the lattice of partial clones is uncountable; nonetheless, Schnoor and Schnoor show that for such algorithmic problems as enumeration, equivalence, and many others, a countable spine of this lattice can be identified that is good enough for obtaining a complexity-theoretic classification.

The interesting issue of the complexity of approximation for optimization problems makes also sense in the context of CSPs. The survey by P. Jonsson and G. Nordh introduces an optimization variant of the constraint satisfaction problem, and presents complexity and approximability results. This variant, which associates a weight to every solution, captures many well-known combinatorial optimization problems. Thus, many different problems can be given a uniform treatment when it comes to solve them or to analyze their complexity.

Formulating algorithmic problems as propositional satisfiability (SAT) problems has become an important problem-solving technique that competes with the CSP approach. In particular, there is a considerable interest in transferring SAT techniques to the CSP context. In the last contribution to this volume, O. Kullmann presents an overview of current paradigms of SAT solving.

Boolean Constraint Satisfaction Problems: When Does Post's Lattice Help?

Nadia Creignou[1] and Heribert Vollmer[2]

[1] LIF (CNRS UMR 6166), Aix-Marseille Université, 163 avenue de Luminy, F-13288 Marseille, France
creignou@lif.univ-mrs.fr
[2] Institut für Theoretische Informatik, Leibniz Universität Hannover, Appelstr. 4, D-30167 Hannover, Germany
vollmer@thi.uni-hannover.de

1 Satisfiability Problems

The propositional satisfiability problem SAT, i.e., the problem to decide, given a propositional formula ϕ (without loss of generality in conjunctive normal form CNF), if there is an assignment to the variables in ϕ that satisfies ϕ, is the historically first and standard NP-complete problem [Coo71]. However, there are well-known syntactic restrictions for which satisfiability is efficiently decidable, for example if every clause in the CNF formula has at most two literals (2CNF formulas) or if every clause has at most one positive literal (Horn formulas) or at most one negative literal (dual Horn formulas), see [KL99]. To study this phenomenon more generally, we study formulas with "clauses" of arbitrary shapes, i.e., consisting of applying arbitrary relations $R \subseteq \{0,1\}^k$ to (not necessarily distinct) variables $x_1, \dots, x_k$. A *constraint language* Γ is a finite set of such relations. In the rest of this chapter, Γ and Γ' will always denote Boolean constraint languages. A Γ*-formula* is a conjunction of clauses $R(x_1, \dots, x_k)$ as above using only relations R from Γ. The for us central family of algorithmic problems, parameterized by a constraint language Γ, now is the problem to determine satisfiability of a given Γ-formula, denoted by $\textsc{Csp}(\Gamma)$.

The NP-complete problem 3SAT, the satisfiability problem for CNF formulas with exactly three literals per clause, now is the problem $\text{CSP}(\Gamma_{3\text{SAT}})$, where $\Gamma_{3\text{SAT}} = \{x \vee y \vee z, x \vee y \vee \neg z, x \vee \neg y \vee \neg z, \neg x \vee \neg y \vee \neg z\}$; here and in the sequel we do not distinguish between a formula ϕ and the logical relation R_ϕ it defines, i.e., the relation consisting of all satisfying assignments of ϕ. If every relation in Γ is definable by a Horn formula, then $\textsc{Csp}(\Gamma)$ is polynomial-time decidable, also if every relation in Γ is definable by a 2-CNF formula. Hence we see that the family of problems $\textsc{Csp}(\Gamma)$ has NP-complete members as well as easily solvable members.

A question attacked by Thomas Schaefer [Sch78] is the following: Can we determine for each constraint language Γ the complexity of $\textsc{Csp}(\Gamma)$? Is there even a simple algorithm that, given Γ, determines the complexity of $\textsc{Csp}(\Gamma)$? Are there more cases than NP-complete and polynomial-time solvable? In this chapter we will present a way to answer these questions that relies on notions

N. Creignou et al. (Eds.): Complexity of Constraints, LNCS 5250, pp. 3–37, 2008.

and results from universal algebra. A central rôle in our development will be an exploitation of the structure of *Post's lattice* of all Boolean clones, all classes of Boolean functions closed under superposition (composition). Post's lattice will turn out to be a very helpful tool to classify the complexity of $\text{CSP}(\Gamma)$, but also of related algorithmic problems for Γ-formulas and generalizations thereof such as quantified Boolean formulas, for instance counting the number of satisfying assignments of quantified Boolean formulas, model checking for circumscription (minimal satisfiability), and many more.

In this chapter we will first present a full account of Schaefer's Theorem and related results for quantified formulas. Then we will survey complexity classifications obtained for many further computational problems for Boolean constraint satisfaction problems in the recent past, with a particular emphasis on the question when Post's lattice can be used in obtaining the classification and when not.

2 Background from Universal Algebra

A *logical relation* (or *constraint relation*) of arity k is a relation $R \subseteq \{0,1\}^k$. A *constraint* (or *constraint application*) is a formula $R(x_1, \ldots, k_k)$, where R is a logical relation of arity k and the $x_1, \ldots, x_k$ are (not necessarily distinct) variables. An assignment I of truth values to the variables *satisfies* the constraint if $\big(I(x_1), \ldots, I(x_k)\big) \in R$. A *constraint language* Γ is a finite set of logical relations. A *Γ-formula* is a conjunction of constraint applications using only logical relations from Γ. Such a formula ϕ is satisfied by an assignment I if I satisfies all constraints in ϕ simultaneously.

Problem:	$\text{CSP}(\Gamma)$
Input:	a Γ-formula ϕ
Question:	Is ϕ satisfiable, i.e., is there an assignment that satisfies ϕ?

When we want to determine the complexity of all CSP-problems, we will certainly need a way to compare the complexity of $\text{CSP}(\Gamma)$ and $\text{CSP}(\Gamma')$ for different constraint languages Γ and Γ'. For example, to show that some $\text{CSP}(\Gamma)$ is NP-complete we might show that using Γ we can " simulate" or "implement" all relations in $\Gamma_{3\text{SAT}}$, and to show that $\text{CSP}(\Gamma)$ is polynomial-time decidable we might implement all relations in Γ using Horn-formulas. As it turns out, a useful notion of implementation comes from universal algebra, from clone theory.

Definition 2.1. For a constraint language Γ, let $\langle\Gamma\rangle$, the *relational clone* (or *co-clone*) generated by Γ, be the smallest set of relations such that

- $\langle\Gamma\rangle$ contains the equality relation and all relations in Γ, and
- $\langle\Gamma\rangle$ is closed under primitive positive definitions, i.e., if ϕ is a $\langle\Gamma\rangle$-formula and $R(x_1, \ldots, x_n) \equiv \exists y_1 \ldots y_\ell\ \phi(x_1, \ldots, x_n, y_1, \ldots, y_\ell)$, then $R \in \langle\Gamma\rangle$.

Intuitively, $\langle\Gamma\rangle$ contains all relations that can be implemented by Γ and is thus called the *expressive power* of Γ, as justified by the following observation:

Proposition 2.2. *If* $\Gamma \subseteq \langle \Gamma' \rangle$ *then* $\mathrm{CSP}(\Gamma) \leq_m^{\log} \mathrm{CSP}(\Gamma')$.

Proof. Let ϕ be a Γ-formula. We construct a formula ϕ' by performing the following steps:

- Replace every constraint from Γ by its defining existentially quantified $(\Gamma' \cup \{=\})$-formula.
- Delete existential quantifiers.
- Delete equality clauses and replace all variables that are connected via a chain of equality constraints by a common new variable.

Then, obviously, ϕ' is a Γ'-formula, and moreover, ϕ is satisfiable if and only if ϕ' is satisfiable. The complexity of the above transformation is dominated by the last step, which is essentially an instance of the undirected graph reachability problem, which is solvable in logarithmic space [Rei05]. Hence we conclude that $\mathrm{CSP}(\Gamma)$ is reducible to $\mathrm{CSP}(\Gamma')$ under logspace reductions. □

In particular, we thus have shown that the complexity of $\mathrm{CSP}(\Gamma)$ depends only on $\langle \Gamma \rangle$ in the following sense:

Proposition 2.3. *If* $\langle \Gamma \rangle = \langle \Gamma' \rangle$, *then* $\mathrm{CSP}(\Gamma) \equiv_m^{\log} \mathrm{CSP}(\Gamma')$,

Thus, we "only" have to study co-clones in order to obtain a full classification, and the question arises what co-clones there are. Astonishingly, all co-clones, each with a "simple" basis, are known. The key to obtain this list is to study closure properties of relations.

Definition 2.4. Let $f\colon \{0,1\}^m \to \{0,1\}$ and $R \subseteq \{0,1\}^n$. We say that f *preserves* R, $f \approx R$, if for all $x_1, \ldots, x_m \in R$, where $x_i = (x_i[1], x_i[2], \ldots, x_i[n])$, we have

$$\Big(f\big(x_1[1], \cdots, x_m[1]\big), f\big(x_1[2], \cdots, x_m[2]\big), \ldots, f\big(x_1[n], \cdots, x_m[n]\big)\Big) \in R.$$

In other words, $f \approx R$ if the coordinate-wise application of f to a sequence of m vectors in R always results in a vector that again is in R. Then we also say that R is *invariant* under f or that f is a *polymorphism* of R, and for a set of relations Γ we write $\mathrm{Pol}(\Gamma)$ to denote the set of all polymorphisms of Γ, i.e., the set of all Boolean functions that preserve every relation in Γ. For technical reasons, we will exclude the empty relation (constraint) and nullary polymorphisms in the rest of this paper.

It is now straightforward to verify that for every Γ, $\mathrm{Pol}(\Gamma)$ is a *clone*, i.e., a set of Boolean functions that contains all projections (all functions $\mathrm{I}_k^n(x_1, \ldots, x_n) = x_k$ for $1 \leq k \leq n$) and is closed under composition; the smallest clone containing a set B of Boolean functions will be denoted by $[B]$ in the sequel (B is also called a *basis* for $[B]$). In fact, the connection between clones and relational clones is much tighter. For a set B of Boolean functions, let $\mathrm{Inv}(B)$ denote the set of all *invariants* of B, i.e., the set of all Boolean relations that are preserved by every function in B. It can be observed that each $\mathrm{Inv}(B)$ is a relational clone.

As shown first in [Gei68, BKKR69] (see also [Lau06, Sect. 2.9]), the operators Pol-Inv constitute a Galois correspondence between the lattice of sets of Boolean relations and the lattice of sets of Boolean functions. In particular, for every set Γ of Boolean relations and every set B be of Boolean functions,

Proposition 2.5. – $\mathrm{Inv}\big(\mathrm{Pol}(\Gamma)\big) = \langle\Gamma\rangle$,
– $\mathrm{Pol}\big(\mathrm{Inv}(B)\big) = [B]$.

Hence, Proposition 2.2 can equivalently be stated as follows:

Proposition 2.6. *If* $\mathrm{Pol}(\Gamma) \supseteq \mathrm{Pol}(\Gamma')$ *then* $\mathrm{CSP}(\Gamma) \leq_m^{\log} \mathrm{CSP}(\Gamma')$.

Thus, there is a one-to-one correspondence between clones and co-clones and we may compile a full list of relational clones from the list of clones obtained by Emil Post in [Pos20, Pos41]. In these papers, Post presented a complete list of Boolean clones, the inclusion structure among them (see Fig. 1), and a finite basis for each of them (Fig. 2). We do not have enough space here to give a full account of *Post's lattice*, as the structure became known, but we refer the interested reader to [Pip97] for a gentle introduction to clones, co-clones, the Galois connection, and Post's results. A rigorous comprehensive study is given in [Lau06]. Complexity-theoretic applications of Post's lattice in the constraint context but also the Boolean circuit context are surveyed in [BCRV03, BCRV04]. A compilation of all co-clones with simple bases is given in [BRSV05].

For the purpose of this paper, we define the clones by simply giving a basis for each of them, see Fig. 2, i.e., the third column of the table gives for each clone its defining basis. One function appearing in the bases that is maybe not so familiar is the threshold function T_k^n, where $\mathrm{T}_k^n(x_1,\ldots,x_n) = 1 \iff \sum_{i=1}^n x_i \geq k$. Also, for a function f, $\mathrm{dual}(f)$, the dual function of f, is given by $\mathrm{dual}\big(f(a_1,\ldots,a_n)\big) = \overline{f(\overline{a_1},\ldots,\overline{a_n})}$.

Let us turn in a little bit more detail to that part of the lattice that will be important here. First, let us give an additional definition. Assuming a canonical order on variables, one can regard assignments as tuples. Thus, with each quantifier free propositional formula ϕ one can associate the relation R_ϕ of all satisfying assignments of ϕ. In the following, we say that a relation R is *defined by* a formula ϕ if $R = R_\phi$. The clone generated by the logical AND function is denoted by E_2. A relation is preserved by AND if and only if it is Horn, that is, definable by a Horn-formula, i.e., $\mathrm{Inv}(\mathrm{E}_2)$ is the set of all Horn relations. Similarly, $\mathrm{V}_2 = [\{\mathrm{OR}\}]$, and $\mathrm{Inv}(\mathrm{V}_2)$ is the set of all dual Horn relations. Relations definable by 2CNF formulas, the so-called *bijunctive* relations, are exactly those in $\mathrm{Inv}(\mathrm{D}_2)$, where D_2 is the clone generated by the 3-ary majority function. Finally, the clone L_2 is generated by the 3-ary exclusive-or $x \oplus y \oplus z$ (the 3-ary addition in GF[2]), and $\mathrm{Inv}(\mathrm{L}_2)$ is the set of all *affine* formulas, i.e., conjunctions of XOR-clauses (consisting of an XOR of some variables plus maybe the constant 1)—these formulas may also be seen as systems of linear equations over GF[2].

Let us say that a constraint language is *Schaefer*, if it belongs to one of the above four types, i.e., Γ is Horn (i.e., every relation in Γ is Horn), dual Horn, bijunctive, or affine. If Γ is Schaefer then $\mathrm{CSP}(\Gamma)$ is polynomial-time

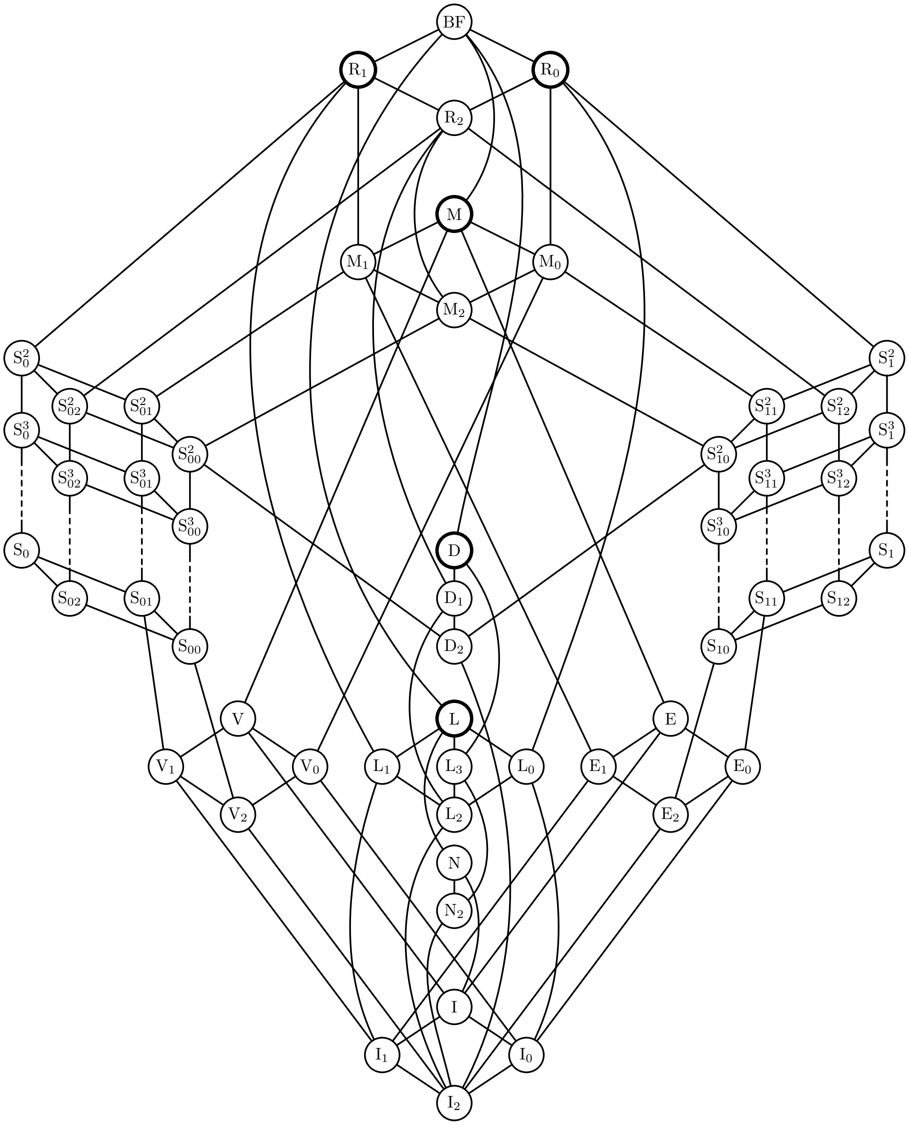


Fig. 1. Post's lattice

solvable, as already noted above for the cases Horn, dual Horn, and bijunctive; for the remaining case of affine relations we remark that we use the interpretation as equations over GF[2] and thus may check satisfiability efficiently using the Gaussian algorithm. (A detailed exposition can be found in [CKS01].)

There is a unique minimal relational clone that is not Schaefer: this is the co-clone Inv(N), where the clone N is generated by the negation function NOT plus the Boolean constants 0, 1. This relational clone consists of all relations

Class	Description	Base
BF	all Boolean functions	$\{\wedge, \neg\}$
R_0	0-reproducing functions	$\{\wedge, \oplus\}$
R_1	1-reproducing functions	$\{\vee, x \oplus y \oplus 1\}$
R_2	$R_1 \cap R_0$	$\{\vee, x \wedge (y \oplus z \oplus 1)\}$
M	monotone functions	$\{\wedge, \vee, 0, 1\}$
M_1	$M \cap R_1$	$\{\wedge, \vee, 1\}$
M_0	$M \cap R_0$	$\{\wedge, \vee, 0\}$
M_2	$M \cap R_2$	$\{\wedge, \vee\}$
S_0^n	functions that are 0-separating of degree n	$\{\rightarrow, \mathrm{dual}(T_n^{n+1})\}$
S_0	0-separating functions	$\{\rightarrow\}$
S_1^n	functions that are 1-separating of degree n	$\{x \wedge \overline{y},\ T_n^{n+1}\}$
S_1	1-separating functions	$\{x \wedge \overline{y}\}$
S_{02}^n	$S_0^n \cap R_2$	$\{x \vee (y \wedge \overline{z}), \mathrm{dual}(T_n^{n+1})\}$
S_{02}	$S_0 \cap R_2$	$\{x \vee (y \wedge \overline{z})\}$
S_{01}^n	$S_0^n \cap M$	$\{\mathrm{dual}(T_n^{n+1}), 1\}$
S_{01}	$S_0 \cap M$	$\{x \vee (y \wedge z), 1\}$
S_{00}^n	$S_0^n \cap R_2 \cap M$	$\{x \vee (y \wedge z), \mathrm{dual}(T_n^{n+1})\}$
S_{00}	$S_0 \cap R_2 \cap M$	$\{x \vee (y \wedge z)\}$
S_{12}^n	$S_1^n \cap R_2$	$\{x \wedge (y \vee \overline{z}), T_n^{n+1}\}$
S_{12}	$S_1 \cap R_2$	$\{x \wedge (y \vee \overline{z})\}$
S_{11}^n	$S_1^n \cap M$	$\{T_n^{n+1}, 0\}$
S_{11}	$S_1 \cap M$	$\{x \wedge (y \vee z), 0\}$
S_{10}^n	$S_1^n \cap R_2 \cap M$	$\{x \wedge (y \vee z), T_n^{n+1}\}$
S_{10}	$S_1 \cap R_2 \cap M$	$\{x \wedge (y \vee z)\}$
D	self-dual functions	$\{x\overline{y} \vee x\overline{z} \vee \overline{yz}\}$
D_1	$D \cap R_2$	$\{xy \vee x\overline{z} \vee y\overline{z}\}$
D_2	$D \cap M$	$\{T_2^3\}$
L	linear functions	$\{\oplus, 1\}$
L_0	$L \cap R_0$	$\{\oplus\}$
L_1	$L \cap R_1$	$\{\equiv\}$
L_2	$L \cap R_2$	$\{x \oplus y \oplus z\}$
L_3	$L \cap D$	$\{x \oplus y \oplus z \oplus 1\}$
V	$\vee$-functions plus constant functions	$\{\vee, 0, 1\}$
V_0	$[\{\vee\}] \cup [\{0\}]$	$\{\vee, 0\}$
V_1	$[\{\vee\}] \cup [\{1\}]$	$\{\vee, 1\}$
V_2	$[\{\vee\}]$	$\{\vee\}$
E	$\wedge$-functions plus constant functions	$\{\wedge, 0, 1\}$
E_0	$[\{\wedge\}] \cup [\{0\}]$	$\{\wedge, 0\}$
E_1	$[\{\wedge\}] \cup [\{1\}]$	$\{\wedge, 1\}$
E_2	$[\{\wedge\}]$	$\{\wedge\}$
N	$[\{\neg\}] \cup [\{0\}] \cup [\{1\}]$	$\{\neg, 1\}, \{\neg, 0\}$
N_2	$[\{\neg\}]$	$\{\neg\}$
I	$I_2 \cup [\{1\}] \cup [\{0\}]$	$\{0, 1\}$
I_0	$I_2 \cup [\{0\}]$	$\{0\}$
I_1	$I_2 \cup [\{1\}]$	$\{1\}$
I_2	all projections	$\emptyset$

Fig. 2. List of all Boolean clones with their defining bases

$\mathrm{Pol}(R) \supseteq \mathrm{V}_2$	$\Leftrightarrow$	R is dual Horn	$\mathrm{Pol}(R) \supseteq \mathrm{E}_2$	$\Leftrightarrow$	R is Horn
$\mathrm{Pol}(R) \supseteq \mathrm{V}_0$	$\Leftrightarrow$	R is definite dual Horn	$\mathrm{Pol}(R) \supseteq \mathrm{E}_1$	$\Leftrightarrow$	R is definite Horn
$\mathrm{Pol}(R) \supseteq \mathrm{S}_{00}$	$\Leftrightarrow$	R is IHSB+	$\mathrm{Pol}(R) \supseteq \mathrm{S}_{10}$	$\Leftrightarrow$	R is IHSB−
$\mathrm{Pol}(R) \supseteq \mathrm{L}_2$	$\Leftrightarrow$	R is affine	$\mathrm{Pol}(R) \supseteq \mathrm{D}_2$	$\Leftrightarrow$	R is bijunctive
$\mathrm{Pol}(R) \supseteq \mathrm{D}_1$	$\Leftrightarrow$	R is affine with width 2	$\mathrm{Pol}(R) \supseteq \mathrm{N}_2$	$\Leftrightarrow$	R is complementive
$\mathrm{Pol}(R) \supseteq \mathrm{I}_1$	$\Leftrightarrow$	R is 1-valid	$\mathrm{Pol}(R) \supseteq \mathrm{I}_0$	$\Leftrightarrow$	R is 0-valid
$\mathrm{Pol}(R) \supseteq \mathrm{I}_2$	$\Leftrightarrow$	R is any relation	$\mathrm{Pol}(R) \supseteq \mathrm{N}$	$\Leftrightarrow$	R is compl., 0- and 1-valid

Fig. 3. Characterizations of some classes of relations

that are at the same time *complementive* (negating all entries of a tuple in the relations leads again to a tuple in the relation), *1-valid* (the all-1 tuple is in the relation), and *0-valid* (the all-0 tuple is in the relation). Because of these latter two properties, satisfiability for CSPs build using only relations from Inv(N) is again efficiently decidable (in fact, they are all satisfiable). If we drop the requirement 1-valid and 0-valid we arrive at the relational clone $\mathrm{Inv}(\mathrm{N}_2)$ consisting of all complementive relations ($\mathrm{N}_2 = [\{\mathrm{NOT}\}]$). Obviously, $\mathrm{Inv}(\mathrm{N}) \subseteq \mathrm{Inv}(\mathrm{N}_2)$, and from Post's lattice it can be seen that there is no relational clone in between. The only super-co-clone of $\mathrm{Inv}(\mathrm{N}_2)$ is the co-clone $\mathrm{Inv}(\mathrm{I}_2)$ of all relations ($\mathrm{I}_2 = [\emptyset]$). These remarks are summarized in Fig. 3.

For some of the classifications we give below, we want to introduce other classes of relations. Let us first introduce classes of formulas which form subclasses of and are less expressive than the class of Horn and dual Horn formulas, namely definite Horn and IHSB (for implicative hitting set bounded); for more background the reader is asked to consult [CKS01]. A *definite Horn* (resp. *definite dual Horn*) formula is a CNF formula having exactly one positive (resp., negative) literal in each clause. A clause is said to be IHSB− if it is of one of the following types: (x_i), $(\neg x_{i_1} \vee x_{i_2})$ or $(\neg x_{i_1} \vee \ldots \vee \neg x_{i_k})$ for some $k \geq 1$. Dually, a clause is said to be IHSB+ if it is of one of the following types: $(\neg x_i)$, $(\neg x_{i_1} \vee x_{i_2})$ or $(x_{i_1} \vee \ldots \vee x_{i_k})$ for some $k \geq 1$. Finally, a formula is said to be IHSB− (resp. IHSB+) if all its clauses are IHSB− (resp. IHSB+).

As usual a Boolean relation R is said to be IHSB− (resp. IHSB+, definite Horn, definite dual Horn) if R can be defined by a CNF formula which is IHSB− (resp. IHSB+, , definite Horn, definite dual Horn). A relation is *affine with width 2* if it is definable by a conjunction of clauses, each of which being either a unary clause or a 2-XOR-clause (consisting of an XOR of 2 variables plus maybe the constant 1)— such a conjunctive formula may also be seen as a system of linear equations over GF[2] with at most two variables per equation. It can be proved that a relation is affine with width 2 if and only if it is both affine and bijunctive. Finally, a constraint language Γ is said to be affine with width 2 (resp. IHSB−, IHSB+, definite Horn, definite dual Horn) if every relation in Γ is affine with width 2 (resp. IHSB−, IHSB+, definite Horn, definite dual Horn).

As for the above introduced classes, all these subclasses of affine and (dual) Horn relations can be characterized by their polymorphisms, see again Fig. 3.

3 Complexity of Satisfiability for Γ-Formulas and Quantified Γ-Formulas

We have seen that if Γ is Schaefer or 0-valid or 1-valid then $\mathrm{CSP}(\Gamma)$ is decidable in polynomial time. If Γ is not of this form, then we have seen that that $\langle\Gamma\rangle \supseteq \mathrm{Inv}(\mathrm{N}_2)$, the co-clone of all complementive relations. A particular example here is the relation

$$\mathrm{R}_{\mathrm{NAE}} = \big\{(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0)\big\}$$

("NAE" here stands for "not all equal"). The language $\mathrm{CSP}(\{\mathrm{R}_{\mathrm{NAE}}\})$ thus consists of 3CNF formulas with only positive literals where we require that in every clause not all literals obtain the same truth value. This is the so-called NOT-ALL-EQUAL problem, known to be NP-complete (see, e.g., [Pap94]). Thus, we have proved *Schaefer's Theorem*:

Theorem 3.1. *If* $\langle\Gamma\rangle \supseteq \mathrm{Inv}(N_2)$ *then* $\mathrm{CSP}(\Gamma)$ *is* NP*-complete, in all other cases,* $\mathrm{CSP}(\Gamma)$ *is polynomial-time decidable.*

In making use of Post's lattice (Fig. 1) this theorem can be reformulated as follows: If $\mathrm{Pol}(\Gamma) \supseteq \mathrm{E}_2$ or $\mathrm{Pol}(\Gamma) \supseteq \mathrm{V}_2$ or $\mathrm{Pol}(\Gamma) \supseteq \mathrm{D}_2$ or $\mathrm{Pol}(\Gamma) \supseteq \mathrm{L}_2$, or if $\mathrm{Pol}(\Gamma) \supseteq \mathrm{I}_0$ or $\mathrm{Pol}(\Gamma) \supseteq \mathrm{I}_1$ then $\mathrm{CSP}(\Gamma)$ is polynomial-time decidable, otherwise $\mathrm{CSP}(\Gamma)$ is NP-complete. Finally, in using the characterizations summarized in Fig. 3, Schaefer's theorem can be stated in more familiar terms: If Γ is Schaefer or 0-valid or 1-valid then $\mathrm{CSP}(\Gamma)$ is polynomial-time decidable, otherwise Γ can express all complementive relations and $\mathrm{CSP}(\Gamma)$ is NP-complete. In the list of complexity results that we will present in Sect. 6 below we will most of the time prefer the formulation as in Theorem 3.1; only in a few particular cases we will additionally present the classification using classical terms.

Because each member of the infinite family of the CSP-problems falls in two complexity cases and avoids the (under the assumption $\mathrm{P} \neq \mathrm{NP}$) infinitely many intermediate degrees, this theorem is also known as Schaefer's *Dichotomy Theorem.*

Recently there has been growing interest in quantified constraints, and we want to survey some of the developments here. The $\mathrm{CSP}(\Gamma)$ problem is equivalent to asking if a Γ-formula with all variables existentially quantified evaluates to true. In the quantified CSP problem one allows also universal quantifiers.

Let us first go one step back and look at usual propositional formulas again. The problem QBF of deciding, whether a given closed quantified propositional formula is true, is PSPACE-complete [SM73], even if the formula is restricted to 3CNF. If the number of quantifier alternations is bounded, the problem is complete in the *polynomial-time hierarchy*, which was defined by Meyer and Stockmeyer [MS72]. Following the notation of [Pap94], $\Sigma_0\mathrm{P} = \Pi_0\mathrm{P} = \mathrm{P}$ and for all $i \geq 0$, $\Sigma_{i+1}\mathrm{P} = \mathrm{NP}^{\Sigma_i\mathrm{P}}$ and $\Pi_{i+1}\mathrm{P} = \mathrm{coNP}^{\Sigma_i\mathrm{P}}$. The set QBF_k of all closed, true quantified Boolean formulas with $k-1$ quantifier alternations starting with an $\exists$-quantifier, is complete for $\Sigma_k\mathrm{P}$ for all $k \geq 1$ [SM73]. This problem remains $\Sigma_k\mathrm{P}$-complete if we restrict the Boolean formula to be 3CNF for k odd, and 3DNF for k even [Wra77]. Since disjunctive normal forms cannot be naturally modelled in a constraint satisfaction context, in order to generalize QBF_k to arbitrary set of constraints Γ in the same way we generalized SAT to $\mathrm{Csp}(\Gamma)$, we consider the unsatisfiability problem for these cases and we adopt the following definition for $\mathrm{Qcsp}_k(\Gamma)$ from [Hem04].

Let Γ be a constraint language and $k \geq 1$. For k odd, a $\mathrm{Qcsp}_k(\Gamma)$ formula is a closed formula of the form $\phi = \exists X_1 \forall X_2 \ldots \exists X_k \psi$, and for k even, a $\mathrm{Qcsp}_k(\Gamma)$ formula is a closed formula of the form $\phi = \forall X_1 \exists X_2 \ldots \exists X_k \psi$, where the X_j, $j = 1, \ldots, k$, are disjoint sets of variables and ψ is a quantifier-free Γ-formula with variables from $\bigcup_j X_j$.

Problem:	$\mathrm{Qcsp}_k(\Gamma)$
Input:	a $\mathrm{Qcsp}_k(\Gamma)$-formula ϕ
Question:	If k is odd: Is ϕ true?
	If k is even: Is ϕ false?

As in the case of simple CSPs above, we note that the Galois connection still helps to study the complexity of QCSP:

Proposition 3.2. *If $\Gamma \subseteq \langle \Gamma' \rangle$ then $\mathrm{Qcsp}_k(\Gamma) \leq_m^{\log} \mathrm{Qcsp}_k(\Gamma')$ for all $k \geq 1$.*

Proof. The proof of this is very similar to the one for Proposition 2.2: Given a Γ-formula ϕ, we construct a formula ϕ' by replacing every constraint from Γ by its defining existentially quantified $\left(\Gamma' \cup \{=\}\right)$-formula. The newly introduced quantified variables will be quantified in the final quantifier block which is by definition of $\mathrm{Qcsp}_k(\Gamma)$-formulas always existential. All that remains to do now is to delete equality clauses as above. □

Certainly, for every constraint language Γ and every $k \geq 1$, $\mathrm{Qcsp}_k(\Gamma) \in \Sigma_k\mathrm{P}$ and $\mathrm{Qcsp}_k(\Gamma_{3\mathrm{SAT}})$ is $\Sigma_k\mathrm{P}$-complete. In fact, even for the single constraint relation

$$\mathrm{R}_{1\text{-IN-}3} = \{(1,0,0),(0,1,0),(0,0,1)\}$$

we have that $\mathrm{Qcsp}_k(\{\mathrm{R}_{1\text{-IN-}3}\})$ is $\Sigma_k\mathrm{P}$-complete. This follows since $\mathrm{R}_{1\text{-IN-}3}$ is only closed under projections and, thus, $\mathrm{Pol}(\mathrm{R}_{1\text{-IN-}3})$ is the minimal clone I_2 in Post's lattice and $\langle \mathrm{R}_{1\text{-IN-}3} \rangle$ is the co-clone $\mathrm{Inv}(\mathrm{I}_2)$ of all Boolean relations. Thus, from Proposition 3.2 we conclude $\mathrm{Qcsp}_k(\Gamma_{3\mathrm{SAT}}) \leq_m^{\log} \mathrm{Qcsp}_k(\{\mathrm{R}_{1\text{-IN-}3}\})$.

In the case of Schaefer's theorem for CSP, already the constraint language consisting of the relation $\mathrm{R}_{\mathrm{NAE}}$ is hard. We want to show an analogous result for QCSP next. To show this, we will reduce $\mathrm{Qcsp}_k(\{\mathrm{R}_{1\text{-IN-}3}\})$ to $\mathrm{Qcsp}_k(\{\mathrm{R}_{\mathrm{NAE}}\})$:

Let ϕ be a $\mathrm{Qcsp}_k(\{\mathrm{R}_{1\text{-IN-}3}\})$-formula,

$$\phi = Q_1 X_1 \ldots \exists X_k \bigwedge_{j=1}^{p} \mathrm{R}_{1\text{-IN-}3}(x_{j_1}, x_{j_2}, x_{j_3}),$$

where Q is existential if k is odd and universal if k is even. We now replace each constraint $\mathrm{R}_{1\text{-IN-}3}(x_{j_1}, x_{j_2}, x_{j_3})$ by the following conjunction:

$$\bigwedge_{j \neq k \in \{j_1, j_2, j_3\}} \mathrm{R}_{\mathrm{NAE}}(x_j, x_k, t) \wedge \mathrm{R}_{\mathrm{NAE}}(x_{j_1}, x_{j_2}, x_{j_3}).$$

It can be checked that this conjunction is true if and only if exactly two of the four variables $x_{j_1}, x_{j_2}, x_{j_3}, t$ are true, hence we will abbreviate the above formula by $\mathrm{R}_{\text{2-IN-4}}(x_{j_1}, x_{j_2}, x_{j_3}, t)$. Now let $\phi' = Q_1 X_1 \dots \exists X_k \bigwedge_{j=1}^{p} \mathrm{R}_{\text{2-IN-4}}(x_{j_1}, x_{j_2}, x_{j_3}, t)$. Since $\mathrm{R}_{\text{1-IN-3}}(x, y, z) = \mathrm{R}_{\text{2-IN-4}}(x, y, z, 1)$, the formula $\phi'[t = 1]$ (every occurrence of t in ϕ is replaced by 1) is true if and only if ϕ is true. Since $\mathrm{R}_{\text{1-IN-3}}(\bar{x}, \bar{y}, \bar{z}) = \mathrm{R}_{\text{2-IN-4}}(x, y, z, 0)$, the formula $\phi'[t/0]$ is true if and only if $\mathrm{Ren}(\phi)$ is true, where $\mathrm{Ren}(\phi)$ is obtained from ϕ by renaming all variables x by their negation $\bar{x}$. Finally, since $\mathrm{Ren}(\phi)$ is true if and only if ϕ is true, we proved that ϕ is true if and only if ϕ' is true. Thus, $\mathrm{QCSP}_k(\{\mathrm{R}_{\text{1-IN-3}}\}) \leq_m^{\log} \mathrm{QCSP}_k(\{\mathrm{R}_{\mathrm{NAE}}\})$.

Hence we now know that if $\langle \Gamma \rangle \supseteq \mathrm{Inv}(\mathrm{N}_2)$, then $\mathrm{QCSP}_k(\Gamma)$ is complete for $\Sigma_k\mathrm{P}$ for every $k \geq 1$. What about the next lower relational clone $\mathrm{Inv}(\mathrm{N})$? In the case of $\mathrm{CSP}(\Gamma)$ (i.e., $\mathrm{QCSP}_1(\Gamma)$), satisfiability is trivial for all $\Gamma \subseteq \mathrm{Inv}(\mathrm{N})$, since every formula is satisfied by the constant-0 or constant-1 assignment. However, this tells us nothing about $\mathrm{QCSP}_k(\Gamma)$ for $k \geq 2$. Let us look at the relation

$$R_0 = \left\{ (u, v, x_1, x_2, x_3) \,\middle|\, u = v \text{ or } \mathrm{R}_{\mathrm{NAE}}(x_1, x_2, x_3) \right\}.$$

It is easy to see that R_0 is complementive, 0-valid, and 1-valid. We will show that $\mathrm{QCSP}_k(\{\mathrm{R}_{\mathrm{NAE}}\})$ reduces to $\mathrm{QCSP}_k(\{R_0\})$. Let

$$\phi = Q_1 X_1 \dots \exists X_k \bigwedge_{j=1}^{p} \mathrm{R}_{\mathrm{NAE}}(x_{j_1}, x_{j_2}, x_{j_3}),$$

where Q_1 is existential if k is odd and universal if k is even, be an instance of $\mathrm{QCSP}_k(\{\mathrm{R}_{\mathrm{NAE}}\})$. We define

$$\phi' = Q_1 X_1 \dots \forall X_{k-1} \forall u \forall v \exists X_k \bigwedge_{j=1}^{p} R_0(u, v, x_{j_1}, x_{j_2}, x_{j_3}).$$

Clearly ϕ is true if and only if ϕ' is true, thus $\mathrm{QCSP}_k(\{\mathrm{R}_{\mathrm{NAE}}\}) \leq_m^{\log} \mathrm{QCSP}_k(\{R_0\})$ for all $k \geq 2$.

We conclude that if $\langle \Gamma \rangle \supseteq \mathrm{Inv}(\mathrm{N})$, then $\mathrm{QCSP}_k(\Gamma)$ is complete for $\Sigma_k\mathrm{P}$ for every $k \geq 2$. If we drop the bound on the number of quantifier alternations and denote the resulting problem by $\mathrm{QCSP}(\Gamma)$, we know from [SM73] that $\mathrm{QCSP}(\Gamma_{3\mathrm{SAT}})$ is PSPACE-complete. The just given reductions thus also show that if $\langle \Gamma \rangle \supseteq \mathrm{Inv}(\mathrm{N})$, then $\mathrm{QCSP}(\Gamma)$ is PSPACE-complete.

If Γ does not include $\mathrm{Inv}(\mathrm{N})$, we know from the structure of Post's lattice that it must be Schaefer. However, it is known that in all four cases (Horn, dual Horn, bijunctive, and affine), the evaluation of quantified formulas is computable in polynomial time (the algorithms for the first three cases rely on Q-resolution, a variant of resolution for quantified propositional formulas, see [KL99]; the algorithm for the affine case is a refinement of the Gaussian algorithm, see [CKS01]). Thus we have proved the following classification:

Theorem 3.3. *If Γ is Schaefer then* $\mathrm{QCSP}(\Gamma)$ *is polynomial-time decidable, in all other cases,* $\mathrm{QCSP}(\Gamma)$ *is* PSPACE-*complete.*

This result was stated without proof and only for constraint languages that include the constants in Schaefer's paper [Sch78]. In its full form it was stated and proven for the first time in [Dal97] and later published in [CKS01].

Looking at QCSPs with bounded quantifier alternations we obtain with the same proof as above *Hemaspaandra's Theorem* [Hem04]. For all $k \geq 2$ (the case $k = 1$ is given by Schaefer's Theorem) the following holds:

Theorem 3.4. *If Γ is Schaefer then* $\text{Q}\textsc{csp}_k(\Gamma)$ *is polynomial-time decidable, in all other cases,* $\text{Q}\textsc{csp}_k(\Gamma)$ *is* $\Sigma_k\text{P}$*-complete.*

Theorems 3.1, 3.3, and 3.4 were originally proven in a different much more involved way in [Sch78, Hem04]. The above simple proofs using Galois theory appeared later. The proof of Theorem 3.1 is implicit in [JCG97, Dal00]. The proofs of Theorems 3.3 and 3.4 are from [BBC+07]. Yet a different proof is given in [Che06].

4 When Does Post's Lattice Help?

Many further results, classifying the computational complexity of different algorithmic tasks for Boolean CSPs have been obtained in the past decades. Some of these rely on the algebraic approach explained above, for others this approach does not seem to be useful. To make this a little bit more precise, let $\Pi(\Gamma)$ be any computational problem defined for Γ-formulas or quantified Γ-formulas. If a result as

$$\text{If } \Gamma \subseteq \langle \Gamma' \rangle \text{then } \Pi(\Gamma) \leq_m^{\log} \Pi(\Gamma') \quad (1)$$

can be proven and then be used to obtain a complexity classification of Π, then we will say that the Galois connection holds *a priori* for Π. For the problems studied in the previous section, the Galois connection holds *a priori.*

For many problems that we will address below, a classification cannot be obtained with the help of a result as (1). Instead, the classification was obtained in sometimes very involved and technically complicated ways making use of different types of implementing one constraint relation by another. However, once the full classification is obtained, it sometimes happens incidentally that it obeys the borders among co-clones, that is, (1) holds but it can only be read from the obtained classification and not be used to obtain the classification. In such a case, we will say that the Galois connection holds *a posteriori* for the problem Π under consideration.

Also, there are some problems where the Galois connection simply does not hold, i.e., an implication as (1) is not true and the known complexity classification does not follow Post's lattice; there might, e.g., exist constraint languages with different complexities that nevertheless give rise to the same co-clone.

The distinction between *a priori* and *a posteriori* should of course not be taken as a mathematical definition—after all, in both cases (1) holds. The distinction should better be regarded as a historical notion.

In Sect. 6 we will survey complexity classifications obtained for many computational problems in the past decades. For each of them, we will pay particular attention to the question if Post's lattice could be used. Before we do so, however, we would like to address an important point concerning the type of reduction that is obtained from the Galois connection.

5 Reducibilities

Schaefer's Dichotomy theorem states that $\mathrm{CSP}(\Gamma)$ is either NP-complete or polynomial-time decidable. This means that under polynomial-time many-one reductions, there are only two possible degrees of complexity for this decision problem.

However, the statement of the Galois connection, Proposition 2.2, speaks of logspace many-one reductions. Hence, together with Post's lattice this can be used to determine all degrees of $\mathrm{CSP}(\Gamma)$ with respect to logspace m-reductions. First results in this direction appear already in Schaefer's paper [Sch78, Theorem 5.1] and in [CKS01, Theorem 6.5]. A thorough examination has been undertaken by Allender, Bauland, Immerman, Schnoor and Vollmer [ABI$^+$05], and it was shown there that the complexity of $\mathrm{CSP}(\Gamma)$ falls into one of five logspace m-degrees: NP-complete, P-complete, NL-complete, $\oplus$L-complete, or decidable in logspace. Here, a logspace m-degree is a class of the form $[A]_{\equiv_m^{\log}} = \{ B \mid B \equiv_m^{\log} A \}$ for some language A, where $B \equiv_m^{\log} A$ denotes that $A \leq_m^{\log} B$ and $B \leq_m^{\log} A$. The name stems from the fact that we are talking about logspace m-reductions $\leq_m^{\log}$.

In fact, Allender *et al.* even make a further step by looking at still stricter reductions, namely the so-called AC^0 many-one-reductions $\leq_m^{\mathrm{AC}^0}$. The class AC^0 consists of all languages/functions computable by uniform families of Boolean circuits of polynomial size and constant depth; for an exact definition and a thorough discussion of the type of uniformity involved we refer the reader to [Vol99]. Now $\leq_m^{\mathrm{AC}^0}$ reductions are just many-one reductions where the reduction function is computable by AC^0 circuits. These reductions are also known as FO-reductions, since the reduction function can be defined by first-order formulas, see [Imm99].

Example 5.1. Let $\Gamma_1 = \{\overline{x}, x\}$. An easy calculation, using Post's lattice, shows that $\mathrm{Pol}(\Gamma_1) = \mathrm{R}_2$, the class of all Boolean functions f that are at the same time 0-reproducing and 1-reproducing, i.e., $f(0,\ldots,0) = 0$ and $f(1,\ldots,1) = 1$. Now, define $\Gamma_2 = \Gamma_1 \cup \{=\}$, then obviously $\mathrm{Pol}(\Gamma_1) = \mathrm{Pol}(\Gamma_2)$.

Formulas over Γ_1 only contain clauses of the form x or $\overline{x}$ for some variables x, such a formula is unsatisfiable if and only if for some variable x, both x and $\overline{x}$ are clauses. This is easily decidable by AC^0 circuits, and $\mathrm{CSP}(\Gamma_1) \in \mathrm{AC}^0$.

In Γ_2 we additionally have the binary equality predicate, and we will now show that $\mathrm{CSP}(\Gamma_2)$ is complete for L under $\leq_m^{\mathrm{AC}^0}$ reductions: The complement of the graph accessibility problem (GAP) for undirected graphs, which is known to be complete for L [Rei05], can be reduced to $\mathrm{CSP}(\Gamma_2)$ as follows: Given a