

# A Computational Trichotomy for Connectivity of Boolean Satisfiability

**Konrad W. Schwerdtfeger**

k.w.s@gmx.net

*Institut für Theoretische Informatik,*

*Leibniz Universität Hannover*

*Appelstr. 4, 30167 Hannover, Germany*

## Abstract

For Boolean satisfiability problems, the structure of the solution space is characterized by the solution graph, where the vertices are the solutions, and two solutions are connected iff they differ in exactly one variable. Motivated by research on heuristics and the satisfiability threshold, in 2006, Gopalan et al. studied connectivity properties of the solution graph and related complexity issues for constraint satisfaction problems [11]. They found dichotomies for the diameter of connected components and for the complexity of the *st*-connectivity question, and conjectured a trichotomy for the connectivity question. Their results could be improved based on findings by Makino et al. [15].

Building on this work, we here prove the trichotomy for the connectivity question. Also, we correct a minor mistake in [11], which leads to a slight shift of the boundaries towards the hard side.

**KEYWORDS:** *Boolean satisfiability, Boolean CSPs, computational complexity, PSPACE-completeness, dichotomy theorems, graph connectivity*

*Submitted January 2014; revised May 2014; published June 2014*

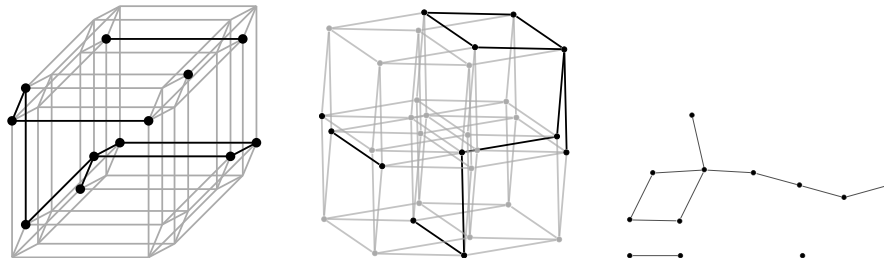
## 1. Introduction

In 2006, P. Gopalan, P. G. Kolaitis, E. Maneva, and C. H. Papadimitriou investigated connectivity properties of the solution space of Boolean constraint satisfaction problems [12, 11]. Their work was motivated inter alia by research on heuristics for satisfiability algorithms and threshold phenomena. Indeed, the solution space connectivity is strongly correlated to the performance of standard satisfiability algorithms like WalkSAT and DPLL on random instances: As one approaches the *satisfiability threshold* (the ratio of constraints to variables at which random  $k$ -CNF-formulas become unsatisfiable for  $k \geq 3$ ) from below, the solution space fractures, and the performance of the algorithms breaks down [17, 16]. These insights mainly came from statistical physics, and lead to the development of the *survey propagation algorithm*, which has much better performance on random instances [16].

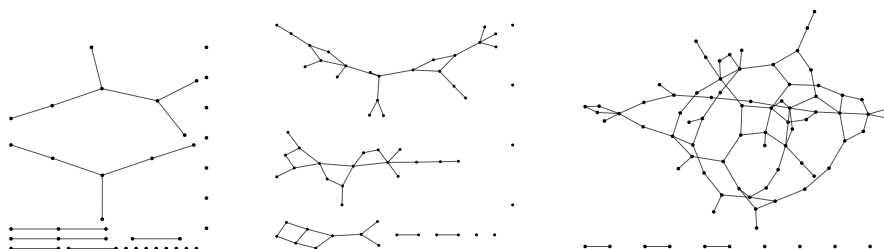
Meanwhile, Gopalan et al.'s results have also been applied directly to *reconfiguration problems*, that arise when a step-by-step transformation between two feasible solutions of a problem is searched, such that all intermediate results are also feasible. Recently, the reconfiguration versions of many problems such as INDEPENDENT-SET, VERTEX-COVER,

SET-COVER GRAPH- $k$ -COLORING, SHORTEST-PATH have been studied [13, 14], and many complexity results were obtained. Another related problem for which the solution space connectivity could be of interest is *structure identification*, where one is given a relation explicitly and seeks a short representation of some kind [8]; this problem is important especially in artificial intelligence.

The solutions (satisfying assignments) of a formula  $\phi$  over  $n$  variables induce a subgraph  $G(\phi)$  of the  $n$ -dimensional hypercube graph, that is, the vertices are the solutions of  $\phi$ , and two solutions are connected iff they differ in exactly one variable.



**Figure 1.** Depictions of the subgraph of the 5-dimensional hypercube graph induced by a typical random Boolean relation with 12 elements. Left: highlighted on a orthographic hypercube projection. Center: highlighted on a “Spectral Embedding” of the hypercube graph by MATHEMATICA.



**Figure 2.** Subgraphs of the 8-dimensional hypercube graph (with 256 vertices) induced by typical random relations with 40, 60 and 80 elements.

Gopalan et al. specifically addressed  $\text{CNF}_C(\mathcal{S})$ -formulas ( $\text{CNF}(\mathcal{S})$ -formulas with constants), see Definition 1, and studied the complexity of the following two decision problems,

- the connectivity problem  $\text{CONN}_C(\mathcal{S})$ , that asks for a given  $\text{CNF}_C(\mathcal{S})$ -formula  $\phi$  whether  $G(\phi)$  is connected,
- the  $st$ -connectivity problem  $\text{ST-CONN}_C(\mathcal{S})$ , that asks for a given  $\text{CNF}_C(\mathcal{S})$ -formula  $\phi$  and two solutions  $s$  and  $t$  whether there a path from  $s$  to  $t$  in  $G(\phi)$ .

Also, they considered

- the maximal diameter of any connected component of  $G(\phi)$  for a  $\text{CNF}_C(\mathcal{S})$ -formula  $\phi$ , where the diameter of a component is the maximal shortest-path distance between any two vectors in that component.

They established a common structural and computational dichotomy, and introduced the corresponding class of *tight* sets of relations, which properly contains all Schaefer sets of relations, see Definition 9: For tight sets  $\mathcal{S}$ , the diameter is linear in the number of variables,  $\text{ST-CONN}_{\mathcal{C}}(\mathcal{S})$  is in P and  $\text{CONN}_{\mathcal{C}}(\mathcal{S})$  is in coNP, while on the other side, the diameter can be exponential, and both problems are PSPACE-complete. Their results are summarized in comparison to the satisfiability problem  $\text{SAT}_{\mathcal{C}}(\mathcal{S})$  in the table below.

**Table 1.** Gopalan et al.’s results [11].

$\mathcal{S}$	$\text{SAT}_{\mathcal{C}}(\mathcal{S})$	$\text{ST-CONN}_{\mathcal{C}}(\mathcal{S})$	$\text{CONN}_{\mathcal{C}}(\mathcal{S})$	Diameter
Schaefer	P	P	coNP	$O(n)$
Tight, not Schaefer	NP-compl.		coNP-compl.	
Not tight			PSPACE-compl.	PSPACE-compl.

Moreover, they conjectured a trichotomy for  $\text{CONN}_{\mathcal{C}}(\mathcal{S})$ : For a certain sub-class of Schaefer sets of relations,  $\text{CONN}_{\mathcal{C}}(\mathcal{S})$  is in P, while for all other tight sets it is coNP-complete.

In Section 3 we will argue that Gopalan et al. did not consider repeated occurrences of variables in constraint applications. As we will see there, repeated variables can make the problems harder and the diameter exponential in some cases, which leads to a slight shift of the boundaries.

In Section 4, we prove the conjectured trichotomy for  $\text{CONN}_{\mathcal{C}}(\mathcal{S})$ , also with the boundaries shifted in the hard direction. Fitted to the correct boundaries, we will introduce the classes of *safely tight* and *CPSS* sets of relations; The supplemental Section 5 will investigate certain properties of CPSS sets of relations. The following table summarizes our results.

**Table 2.** Complete classification of the connectivity problems and the diameter for CNF( $\mathcal{S}$ )-formulas with constants, in comparison to SAT.

$\mathcal{S}$	$\text{SAT}_{\mathcal{C}}(\mathcal{S})$	$\text{ST-CONN}_{\mathcal{C}}(\mathcal{S})$	$\text{CONN}_{\mathcal{C}}(\mathcal{S})$	Diameter
CPSS	P	P	P	$O(n)$
Schaefer, not CPSS			coNP-compl.	
Safely tight, not Schaefer	NP-compl.	PSPACE-compl.	PSPACE-compl.	$2^{\Omega(\sqrt{n})}$
Not safely tight				

## 2. Preliminaries

First we introduce some terminology for Boolean relations and formulas. We will use the standard notions also used in [11], but carefully define *substitution of constants* and *identification of variables*, and distinguish CNF( $\mathcal{S}$ )-formulas with and without constants.

**Definition 1.** An  $n$ -ary Boolean relation (or logical relation, relation for short) is a subset of  $\{0, 1\}^n$  ( $n \geq 1$ ).

For an  $n$ -ary relation  $R$ , we can define an  $(n - k)$ -ary relation

$$R'(x_1, \dots, x_{n-k}) = R(\xi_1, \dots, \xi_n)$$

( $0 < k < n$ ). If each  $\xi_i \in \{0, 1, x_1, \dots, x_{n-k}\}$  and each variable  $x_i \in \{x_1, \dots, x_{n-k}\}$  occurs at most once in  $(\xi_1, \dots, \xi_n)$ , we say  $R'$  is obtained from  $R$  by substitution of constants. If each  $\xi_i \in \{x_1, \dots, x_{n-k}\}$ , (and each  $x_i \in \{x_1, \dots, x_{n-k}\}$  may occur any number of times in  $(\xi_1, \dots, \xi_n)$ ),  $R'$  is obtained by identification of variables. Note that we allow any permutation of the variables in both cases.

The set of solutions of a propositional formula  $\phi$  over  $n$  variables defines in a natural way an  $n$ -ary relation  $[\phi]$ , where the variables are taken in lexicographic order. We will often identify the formula  $\phi$  with the relation it defines and omit the brackets.

In the following definition note that we write  $\text{ST-CONN}_C(\mathcal{S})$  resp.  $\text{CONN}_C(\mathcal{S})$  instead of  $\text{ST-CONN}(\mathcal{S})$  resp.  $\text{ST-CONN}(\mathcal{S})$  like Gopalan et al., for consistency with the usual notation  $\text{SAT}(\mathcal{S})$  for the satisfiability problem without constants and  $\text{SAT}_C(\mathcal{S})$  for the one with constants.

**Definition 2.** A CNF-formula is a propositional formula of the form  $C_1 \wedge \dots \wedge C_m$  ( $1 \leq m < \infty$ ), where each  $C_i$  is a clause, that is, a finite disjunction of literals (variables or negated variables). A  $k$ -CNF-formula ( $k \geq 1$ ) is a CNF-formula where each  $C_i$  has at most  $k$  literals. A Horn (dual Horn) formula is a CNF-formula where each  $C_i$  has at most one positive (negative) literal.

For a finite set of relations  $\mathcal{S}$ , a  $\text{CNF}_C(\mathcal{S})$ -formula over a set of variables  $V$  is a finite conjunction  $C_1 \wedge \dots \wedge C_m$ , where each  $C_i$  is a constraint application (constraint for short), i.e., an expression of the form  $R(\xi_1, \dots, \xi_k)$ , with a  $k$ -ary relation  $R \in \mathcal{S}$ , and each  $\xi_j$  is a variable from  $V$  or one of the constants 0, 1. By  $\text{Var}(C_i)$ , we denote the set of variables occurring in  $\xi_1, \dots, \xi_k$ . With the relation corresponding to  $C_i$  we mean the relation  $[R(\xi_1, \dots, \xi_k)]$  (that may be different from  $R$  by substitution of constants, or identification or permutation of variables). A  $\text{CNF}(\mathcal{S})$ -formula is a  $\text{CNF}_C(\mathcal{S})$ -formula where each  $\xi_j$  is a variable in  $V$ , not a constant.

We define the solution graph and its diameter as in [11]. We use  $\mathbf{a}, \mathbf{b}, \dots$  or  $\mathbf{a}^1, \mathbf{a}^2, \dots$  to denote vectors of Boolean values and  $\mathbf{x}, \mathbf{y}, \dots$  or  $\mathbf{x}^1, \mathbf{x}^2, \dots$  to denote vectors of variables,  $\mathbf{a} = (a_1, a_2, \dots)$  and  $\mathbf{x} = (x_1, x_2, \dots)$ .

**Definition 3.** The solution graph  $G(\phi)$  of  $\phi$  is the subgraph of the  $n$ -dimensional hypercube graph induced by the vectors in  $[\phi]$ , i.e., the vertices of  $G(\phi)$  are the vectors in  $[\phi]$ , and there is an edge between two vectors iff they differ in exactly one variable. We will also refer to  $G(R)$  for any logical relation  $R$  (not necessarily defined by a formula).

If  $\mathbf{a}$  and  $\mathbf{b}$  are solutions of a formula  $\phi$  and lie in the same connected component (component for short) of  $G(\phi)$ , we write  $d_\phi(\mathbf{a}, \mathbf{b})$  to denote the shortest-path distance between  $\mathbf{a}$  and  $\mathbf{b}$ . The diameter of a component is the maximal shortest-path distance between any two vectors in that component. The diameter of  $G(\phi)$  is the maximal diameter of any component.

The Hamming distance  $|\mathbf{a} - \mathbf{b}|$  of two Boolean vectors  $\mathbf{a}$  and  $\mathbf{b}$  is the number of positions in which they differ.

We define the following decision problems for  $\text{CNF}(\mathcal{S})$ -formulas resp.  $\text{CNF}_C(\mathcal{S})$ -formulas:

- the *satisfiability problem*  $\text{SAT}(\mathcal{S})$ : Given a  $\text{CNF}(\mathcal{S})$ -formula  $\phi$ , is  $\phi$  satisfiable?
- the *satisfiability problem with constants*  $\text{SAT}_C(\mathcal{S})$ : Given a  $\text{CNF}_C(\mathcal{S})$ -formula  $\phi$ , is  $\phi$  satisfiable?
- the *connectivity problem (with constants)*  $\text{CONN}_C(\mathcal{S})$ : Given a  $\text{CNF}_C(\mathcal{S})$ -formula  $\phi$ , is  $G(\phi)$  connected? (if  $\phi$  is unsatisfiable, then  $G(\phi)$  is considered connected)
- the *st-connectivity problem (with constants)*  $\text{ST-CONN}_C(\mathcal{S})$ : Given a  $\text{CNF}_C(\mathcal{S})$ -formula  $\phi$  and two solutions  $\mathbf{s}$  and  $\mathbf{t}$ , is there a path from  $\mathbf{s}$  to  $\mathbf{t}$  in  $G(\phi)$ ?

The complexity of the problems depends on the kind of relations in  $\mathcal{S}$ ; we now define the relevant types. Some are already familiar from Schaefer’s classification of SAT, some were introduced by Gopalan et al., and the ones starting with “safely” are new; *IHSB* stands for “implicative hitting set-bounded” and was adopted by Gopalan et al. from [7], where it was introduced for a refinement of Schaefer’s theorem and the classification of related problems.

**Definition 4.** *Let  $R$  be an  $n$ -ary logical relation.*

- $R$  is 0-valid (1-valid) if  $0^n \in R$  ( $1^n \in R$ ).
- $R$  is bijunctive if it is the set of solutions of a 2-CNF-formula.
- $R$  is Horn (dual Horn) if it is the set of solutions of a Horn (dual Horn) formula.
- $R$  is affine if it is the set of solutions of a formula  $x_{i_1} \oplus \dots \oplus x_{i_m} \oplus c$  with  $i_1, \dots, i_m \in \{1, \dots, n\}$  and  $c \in \{0, 1\}$ ,
- $R$  is componentwise bijunctive if every connected component of  $G(R)$  is a bijunctive relation.  $R$  is safely componentwise bijunctive if  $R$  and every relation  $R'$  obtained from  $R$  by identification of variables is componentwise bijunctive.
- $R$  is OR-free (NAND-free) if the relation  $OR = \{01, 10, 11\}$  ( $NAND = \{00, 01, 10\}$ ) cannot be obtained from  $R$  by substitution of constants.  $R$  is safely OR-free (safely NAND-free) if  $R$  and every relation  $R'$  obtained from  $R$  by identification of variables is OR-free (NAND-free).
- $R$  is IHSB– (IHSB+) if it is the set of solutions of a Horn (dual Horn) formula in which all clauses with more than 2 literals have only negative literals (only positive literals).
- $R$  is componentwise IHSB– (componentwise IHSB+) if every connected component of  $G(R)$  is IHSB– (IHSB+).  $R$  is safely componentwise IHSB– (safely componentwise IHSB+) if  $R$  and every relation  $R'$  obtained from  $R$  by identification of variables is componentwise IHSB– (componentwise IHSB+).

If one is given the relation explicitly (as a set of vectors), the properties 0-valid, 1-valid, OR-free and NAND-free can be checked easily. Bijunctive, Horn, dual Horn, affine, IHSB– and IHSB+ can be checked by *closure properties*:

**Definition 5.** A relation  $R$  is closed under some  $n$ -ary operation  $f$  iff the vector obtained by the coordinate-wise application of  $f$  to any  $m$  vectors from  $R$  is again in  $R$ , i.e., if

$$\mathbf{a}^1, \dots, \mathbf{a}^m \in R \implies (f(a_1^1, \dots, a_1^m), \dots, f(a_n^1, \dots, a_n^m)) \in R.$$

**Lemma 6.** A relation  $R$  is

- *bijunctive*, iff it is closed under the ternary majority operation  $MAJ(x, y, z) = (x \vee y) \wedge (y \vee z) \wedge (z \vee x)$  [7, Lemma 4.9],
- *Horn (dual Horn)*, iff it is closed under  $\wedge$  (under  $\vee$ , resp.) [7, Lemma 4.8],
- *affine*, iff it is closed under  $x \oplus y \oplus z$  [7, Lemma 4.10],
- *IHSB− (IHSB+)*, iff it is closed under  $x \wedge (y \vee z)$  (under  $x \vee (y \wedge z)$ , resp.).

*Proof.* For IHSB− and IHSB+ relations, this can be verified using the Galois correspondence between closed sets of relations and closed sets of Boolean functions (see [3]). From the table in [3] we find that the IHSB− relations are a base of the co-clone  $INV(\mathbf{S}_{10})$ , and the IHSB+ ones a base of  $INV(\mathbf{S}_{00})$ , and from the table in [2] we see that  $x \wedge (y \vee z)$  and  $x \vee (y \wedge z)$  are bases of the clones  $\mathbf{S}_{10}$  and  $\mathbf{S}_{00}$ , resp.  $\square$

The following examples show that the “safely” classes are properly contained in the corresponding “unsafe” ones.

**Example 7.** The relation  $\{001, 110, 111\}$  is OR-free, but not safely OR-free, as identifying the first two variables gives  $\{01, 10, 11\}$ .

The smallest examples of relations that are componentwise bijunctive, but not safely componentwise bijunctive, or Horn and componentwise IHSB−, but not safely componentwise IHSB− are of dimension 4:

**Example 8.** For the relation  $R_{\text{coNP}} = \{0000, 0100, 1100, 0011, 1011\}$ , both components  $\{0000, 0100, 1100\}$  and  $\{0011, 1011\}$  are closed under MAJ and under  $x \vee (y \wedge z)$ , but the relation  $R' = \{000, 010, 110, 001, 101\}$ , obtained from  $R_{\text{coNP}}$  by identifying the third and fourth variable, has only one component that is neither closed under MAJ, nor under  $x \wedge (y \vee z)$ : applying MAJ or  $x \wedge (y \vee z)$  coordinate-wise to  $(110, 000, 101)$  both gives  $100 \notin R'$ .

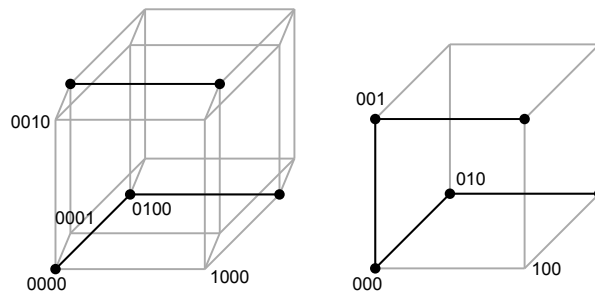
For an example of a formula consider

$$\phi_{\text{coNP}} = (x \wedge y) \vee (\bar{x} \wedge \bar{y} \wedge (\bar{z} \vee \bar{w})) \equiv (x \vee \bar{y}) \wedge (\bar{x} \vee y) \wedge (x \vee \bar{z} \vee \bar{w}) \wedge (y \vee \bar{z} \vee \bar{w}),$$

which is clearly componentwise bijunctive and componentwise IHSB−, but  $x \vee \bar{z} \vee \bar{w}$ , obtained by identifying  $y$  with  $x$ , has only one component that is neither bijunctive nor IHSB−.

The following classes of sets of relations are fitted to the structural and computational boundaries for the connectivity; the term *CPSS* stands for *constraint-projection separating Schaefer* and will become clear in Section 5 from Definition 20 and Lemma 24.

**Definition 9.** A set  $\mathcal{S}$  of logical relations is tight (safely tight) if at least one of the following conditions holds:



**Figure 3.** The solution graphs of the relations  $R_{\text{coNP}}$  and  $R'$  from Example 8, drawn on orthographic hypercube projections; the “axis vertices” are labeled.

1. every relation in  $\mathcal{S}$  is componentwise bijunctive (safely componentwise bijunctive).
2. every relation in  $\mathcal{S}$  is OR-free (safely OR-free).
3. every relation in  $\mathcal{S}$  is NAND-free (safely NAND-free).

A set  $\mathcal{S}$  of logical relations is Schaefer if at least one of the following conditions holds:

1. every relation in  $\mathcal{S}$  is bijunctive.
2. every relation in  $\mathcal{S}$  is Horn.
3. every relation in  $\mathcal{S}$  is dual Horn.
4. every relation in  $\mathcal{S}$  is affine.

A set  $\mathcal{S}$  of logical relations is CPSS if at least one of the following conditions holds:

1. every relation in  $\mathcal{S}$  is bijunctive.
2. every relation in  $\mathcal{S}$  is Horn and safely componentwise IHSB-.
3. every relation in  $\mathcal{S}$  is dual Horn and safely componentwise IHSB+.
4. every relation in  $\mathcal{S}$  is affine.

### 3. The Impact of Repeated Variables in Constraints

An upcoming extended version of this paper will give a self contained account, here we require Gopalan et al.’s definitions and results, and just exactly show how the affected statements and proofs in [11] need to be modified to take into account repeated occurrences of variables in constraint applications.

The central concept of *structural expressibility* (Definition 3.1 in [11]), as well as the PSPACE-completeness proof for the connectivity problems for 3-CNF-formulas (Lemma 3.6 in [11]) is not affected.

The first mistake is in the generalization of the structural properties from  $\text{CNF}_C(\mathcal{S})$ -formulas with bijunctive sets  $\mathcal{S}$  of relations to those with componentwise bijunctive sets  $\mathcal{S}$

in subsection 4.2 of [11], the other mistakes are similar. In the following, we refer to the Lemmas, Theorems and Corollaries of [11].

As the proofs are stated there, the flaws are quite hard to locate. We have to carefully distinguish the relation from  $\mathcal{S}$  used in a constraint of a  $\text{CNF}_{\mathcal{C}}(\mathcal{S})$ -formula as “template” from the resulting relation for the variables of the formula, see Definition 2. The second paragraph of the proof of Lemma 4.3 is supposed to show that every component of a  $\text{CNF}_{\mathcal{C}}(\mathcal{S})$ -formulas  $\varphi$  with a set  $\mathcal{S}$  of only componentwise bijunctive relations is the solution space of a formula  $\varphi'$  with only bijunctive relations. To construct  $\varphi'$ , every constraint  $C_i$  in  $\varphi$  using a relation  $R \in \mathcal{S}$  with more than one component is replaced by a constraint containing only one component of  $R$ .

But if in  $C_i$  some variables of  $R$  are identified, and  $R$  only is componentwise bijunctive and not safely componentwise bijunctive, it is possible that the relation resulting for the variables of  $\varphi$  is not componentwise bijunctive, as we have seen in Example 8, and thus not every of its components is bijunctive.

So we must change the Lemma to

- LEMMA 4.3: *Let  $\mathcal{S}$  be a set of **safely** componentwise bijunctive relations and  $\varphi$  a  $\text{CNF}_{\mathcal{C}}(\mathcal{S})$ -formula. If  $\mathbf{a}$  and  $\mathbf{b}$  are two solutions of  $\varphi$  that lie in the same component of  $G(\varphi)$ , then  $d_{\varphi}(\mathbf{a}, \mathbf{b}) = |\mathbf{a} - \mathbf{b}|$ , i.e., no distance expands.*

In the proof, we replace the second paragraph by

“For the general case, we show that every component  $F$  of  $G(\varphi)$  is the solution space of a 2-CNF-formula  $\varphi$ . Let  $R \in \mathcal{S}$  be a safely componentwise bijunctive relation. Then any relation corresponding to a clause in  $\varphi$  of the form  $R(x_1, \dots, x_k)$  (the relation obtained after identifying repeated variables) consists of bijunctive components  $R_1, \dots, R_m$ . The projection of  $F$  onto  $x_1, \dots, x_k$  is itself connected and must satisfy  $R$ . Hence it lies within one of the components  $R_1, \dots, R_m$ ; assume it is  $R_1$ . We replace  $R(x_1, \dots, x_k)$  by  $R_1(x_1, \dots, x_k)$ . Call this new formula  $\varphi_1$ .  $G(\varphi_1)$  consists of all components of  $G(\varphi)$  whose projection on  $x_1, \dots, x_k$  lies in  $R_1$ . We repeat this for every clause. Finally we are left with a formula  $\varphi'$  over a set of bijunctive relations. Hence  $\varphi'$  is bijunctive and  $G(\varphi')$  is a component of  $G(\varphi)$ . So the claim follows from the bijunctive case.”

In consequence, the resulting Corollary must be changed to

- COROLLARY 4.4: *Let  $\mathcal{S}$  be a set of **safely** componentwise bijunctive relations. Then...*

The proof of Lemma 4.5 is supposed to show by contradiction that every component of  $G(\varphi)$  for a  $\text{CNF}_{\mathcal{C}}(\mathcal{S})$ -formula  $\varphi$  with a set  $\mathcal{S}$  of OR-free relations must contain a unique locally minimal solution. It is reasoned that if  $G(\varphi)$  would contain two locally minimal solutions, the relation corresponding to some clause  $C_i$  in  $\varphi$  would not be OR-free. This is correct up to here, with our Definition 2 of “the relation corresponding to a clause”. From this it is concluded that some relation in  $\mathcal{S}$  could not have been OR-free. But actually,  $C_i$  could have been obtained from an OR-free relation that is not safely OR-free by identification of variables, as we have seen in Example 7. So the Lemma must be changed to

- LEMMA 4.5: *Let  $\mathcal{S}$  be a set of **safely** OR-free relations and  $\varphi$  a  $\text{CNF}_{\mathcal{C}}(\mathcal{S})$ -formula. Every component...*



In the proof, we replace the last sentence of the second paragraph by “So the relation corresponding to that clause is not OR-free, thus  $\mathcal{S}$  must have contained some not safely OR-free relation.”

In consequence, the resulting Corollaries must be changed to

- COROLLARY 4.6: *Let  $\mathcal{S}$  be a set of **safely** OR-free relations. Then...*

and

- COROLLARY 4.7: *Let  $\mathcal{S}$  be a **safely** tight set of relations. Then...*

But now for dichotomies to hold, we must show for every not safely tight set  $\mathcal{S}$  that  $\text{CONN}_C(\mathcal{S})$  and  $\text{ST-CONN}_C(\mathcal{S})$  are PSPACE-complete, and that there are  $\text{CNF}_C(\mathcal{S})$ -formulas  $\varphi$  such that the diameter of  $G(\varphi)$  is exponential in the number of variables of  $\varphi$ . Therefore, we extend the structural expressibility of  $S_3$  to not safely tight sets of relations; we change LEMMA 3.4 to

- LEMMA 3.4: *If set  $\mathcal{S}$  of relations is not **safely** tight,  $S_3$  is structurally expressible from  $\mathcal{S}$ .*

In the first paragraph of the proof, we replace “not OR-free” with “not **safely** OR-free”, “not NAND-free” with “not **safely** NAND-free”, and note that we can express  $x_1 \vee x_2$  ( $\bar{x}_1 \vee \bar{x}_2$ ) by substitution of constants **and identification of variables** from any such set of relations. Similarly, in the first paragraph of “Step 1”, we replace “componentwise bijunctive” by “**safely** componentwise bijunctive”, and in the second paragraph of “Step 1” we obtain the required not componentwise bijunctive relation  $R$  from any not safely componentwise bijunctive relation  $R'$  by identification of variables. The remaining part of the proof need not be modified.

Now we can extend the structural expressibility theorem:

- THEOREM 2.7: *Let  $\mathcal{S}$  be a finite set of logical relations. If  $\mathcal{S}$  is not **safely** tight, then every logical relation is structurally expressible from  $\mathcal{S}$ .*

Hereby, we can state the dichotomy theorems as follows:

- THEOREM 2.8: *Let  $\mathcal{S}$  be a finite set of logical relations. If  $\mathcal{S}$  is **safely** tight, then  $\text{CONN}_C(\mathcal{S})$  is in  $\text{CONP}$ ; otherwise,  $\text{CONN}_C(\mathcal{S})$  is PSPACE-complete.*
- THEOREM 2.9: *Let  $\mathcal{S}$  be a finite set of logical relations. If  $\mathcal{S}$  is **safely** tight, then  $\text{ST-CONN}_C(\mathcal{S})$  is in  $\text{P}$ ; otherwise,  $\text{ST-CONN}_C(\mathcal{S})$  is PSPACE-complete.*
- THEOREM 2.10: *Let  $\mathcal{S}$  be a finite set of logical relations. If  $\mathcal{S}$  is **safely** tight, then for every  $\text{CNF}_C(\mathcal{S})$ -formula  $\varphi$ , the diameter of  $G(\varphi)$  is linear in the number of variables of  $\varphi$ ; otherwise, there are  $\text{CNF}_C(\mathcal{S})$ -formulas  $\varphi$  such that the diameter of  $G(\varphi)$  is exponential in the number of variables of  $\varphi$ .*

For the inclusion structure of the classes to hold, we now have to show that all Schaefer sets of relations are safely tight. Therefore, we tighten LEMMA 4.2 of [11]:

- LEMMA 4.2: *Let  $R$  be a logical relation.*
  1. *If  $R$  is bijunctive, then  $R$  is **safely** componentwise bijunctive*
  2. *If  $R$  is Horn, then  $R$  is **safely** OR-free.*
  3. *If  $R$  is dual Horn, then  $R$  is **safely** NAND-free.*
  4. *If  $R$  is affine, then  $R$  is **safely** componentwise bijunctive, **safely** OR-free, and **safely** NAND-free.*

For the proof, we first note that any relation obtained from a bijunctive (Horn, dual Horn, affine) one by identification of variables is itself bijunctive (Horn, dual Horn, affine), which is obvious from the definitions.

Using this fact, it is clear from the definitions of safely componentwise bijunctive, safely OR-free, and safely NAND-free, that the original proof implies that every bijunctive relation is safely componentwise bijunctive, every Horn (dual Horn) relation is safely OR-free (NAND-free), and every affine relation is safely componentwise bijunctive, safely OR-free, and safely NAND-free.

We have to weaken LEMMA 4.8, since it relies on the wrong assumption that  $\text{CONN}_C(\mathcal{S})$  is in coNP for every tight set  $\mathcal{S}$ :

- LEMMA 4.8: *For  $\mathcal{S}$  **safely** tight, but not Schaefer,  $\text{CONN}_C(\mathcal{S})$  is coNP-complete.*

In the proof, we should clarify that the relation  $x \neq y$  is expressible as a  $\text{CNF}_C(\mathcal{S})$ -formula, not necessarily by substitution of constants only, see Remark 11.

Finally, we have to weaken Lemma 4.13. In the last paragraph of the proof, the connectivity question for a  $\text{CNF}_C(\mathcal{S})$ -formula  $\varphi$  with a set  $\mathcal{S}$  of componentwise IHSB– relations shall be reduced to one for a formula using only IHSB– relations. In the last sentence, a false assumption is used: That every relation corresponding to a clause of  $\varphi$  that has only a single component would be IHSB–. Actually, that relation is guaranteed to be IHSB– only if the original relation is safely componentwise IHSB–, as we have seen in Example 8. Thus the Lemma must be changed to

- LEMMA 4.13: *If  $\mathcal{S}$  a set of relations that are Horn (dual Horn) and **safely** componentwise IHSB– (IHSB+), then there is a polynomial-time algorithm for  $\text{CONN}_C(\mathcal{S})$ .*

The proof can be retained word-for-word; the necessary comment “(the relation obtained after identifying repeated variables)” is already mentioned in the last paragraph.

In the next section, we will proof that if  $\mathcal{S}$  is a finite set of Horn (dual Horn) relations that contains at least one relation that is not safely componentwise IHSB– (IHSB+), then  $\text{CONN}_C(\mathcal{S})$  is coNP-complete (Lemma 13).

The following example shows cases where the above corrections make a difference.

**Example 10.** Since the relations from Example 8 are Horn and componentwise IHSB–,  $\text{CONN}_C(\{R_{\text{coNP}}\})$  and  $\text{CONN}_C(\{\phi_{\text{coNP}}\})$  would be polynomial-time decidable by Lemma 4.13 of [11]. But  $\{R_{\text{coNP}}\}$  and  $\{\phi_{\text{coNP}}\}$  are not CPSS, and consequently,  $\text{CONN}_C(\{R_{\text{coNP}}\})$  and  $\text{CONN}_C(\{\phi_{\text{coNP}}\})$  are actually coNP-complete.

The relation

$$R_{\text{PSPA}} = \{0001, 0010, 1100, 1110, 1101\}$$

is not Schaefer and not NAND-free ( $R_{\text{PSPA}}(1, 1, x, y) = \overline{x \wedge y}$ ), but componentwise bijunctive and OR-free, thus  $\{R_{\text{PSPA}}\}$  is tight, and  $\text{CONN}_C(\{R_{\text{PSPA}}\})$  would be coNP-complete by Lemma 4.8 of [11],  $\text{ST-CONN}_C(\{R_{\text{PSPA}}\})$  would be polynomial-time decidable by Theorem 2.9 of [11], and the diameter of  $G(\phi)$  linear in the number of variables for all  $\text{CNF}_C(\{R_{\text{PSPA}}\})$ -formulas  $\phi$  by Theorem 2.10 of [11].

But  $R_{\text{PSPA}}$  is not safely componentwise bijunctive (as identifying the first two variables gives  $R' = \{001, 010, 100, 110, 101\}$ , and  $\text{MAJ}(001, 010, 100) = 000 \notin R'$ ), and not safely OR-free (as  $R'(x, y, 0) = x \vee y$ ), thus  $\{R_{\text{PSPA}}\}$  is not safely tight, and  $\text{CONN}_C(\{R_{\text{PSPA}}\})$  and  $\text{ST-CONN}_C(\{R_{\text{PSPA}}\})$  are actually PSPACE-complete, and there are  $\text{CNF}_C(\{R_{\text{PSPA}}\})$ -formulas  $\phi$  for which the diameter of  $G(\phi)$  is exponential in the number of variables.

*Remark 11.* One could of course also consider  $\text{CNF}_C(\mathcal{S})$ -formulas without repeated variables in constraints. But in this case, one had to check all consequences of this restriction. E.g., the proof of Lemma 4.8 in [11] were not valid since the relation  $x \neq y$  is not expressible without identification of variables from every non-Schaefer set of relations. For example, for  $R = \{1100, 1010, 1110, 0001\}$ ,  $(x \neq y)$  is  $R(x, x, x, y)$ , but cannot be obtained from  $R$  by substitution of constants and conjunction only.

#### 4. A Trichotomy for $\text{CONN}_C(\mathcal{S})$

In this section, we prove the last piece needed to establish the trichotomy for  $\text{CONN}_C(\mathcal{S})$ .

Initially, Gopalan et al. conjectured that  $\text{CONN}_C(\mathcal{S})$  is in P if  $\mathcal{S}$  is Schaefer, but this was subsequently disproved by Makino, Tamaki, and Yamamoto [15], who showed that  $\text{CONN}_C(\mathcal{S})$  is coNP-complete for  $\mathcal{S} = \{x \vee \bar{y} \vee \bar{z}\}$ , which is Horn and thus Schaefer. Consequently, Gopalan et al. conjectured that  $\text{CONN}_C(\mathcal{S})$  is coNP-complete if  $\mathcal{S}$  is Horn but not componentwise IHSB $-$ , or dual Horn but not componentwise IHSB $+$ , and already suggested a way for proving that: One had to show that  $\text{CONN}_C(\{M\})$  for the relation  $M = (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)$  is coNP-hard. We will prove this in Lemma 18 by a reduction from the complement of a satisfiability problem.

Gopalan et al. stated (without giving the proof) they could show that  $M$  is structurally expressible from every set of Horn relations which contains at least one relation that is not componentwise IHSB $-$ , using a similar reasoning as in the proof of their structural expressibility theorem.

We give a different proof (which may be somewhat simpler) in Lemma 19, that shows that  $M$  actually is expressible from every set  $\mathcal{S}$  of Horn relations that contains at least one relation that is not **safely** componentwise IHSB $-$  as a  $\text{CNF}_C(\mathcal{S})$ -formula, which is of course a structural expression.

In this section, when we use results from [11], we refer to the corrected versions from the last section, where applicable.

**Theorem 12** (Trichotomy theorem for  $\text{CONN}_C(\mathcal{S})$ ). *Let  $\mathcal{S}$  be a finite set of logical relations.*

1. *If  $\mathcal{S}$  is CPSS,  $\text{CONN}_C(\mathcal{S})$  is in P.*
2. *Else if  $\mathcal{S}$  is safely tight,  $\text{CONN}_C(\mathcal{S})$  is coNP-complete.*

3. Else,  $\text{CONN}_C(\mathcal{S})$  is PSPACE-complete.

*Proof.* 1. If  $\mathcal{S}$  is CPSS,  $\text{CONN}_C(\mathcal{S})$  is in P by Lemmas 4.9, 4.13 and 4.10 of [11], or by our Theorem 21.

2. If  $\mathcal{S}$  is Schaefer and not CPSS, it must be Horn and contain at least one relation that is not safely componentwise IHSB $-$ , or dual Horn and contain at least one relation that is not safely componentwise IHSB $+$ , and  $\text{CONN}_C(\mathcal{S})$  is coNP-complete by Lemma 13 below. If  $\mathcal{S}$  is not Schaefer, the statement follows from Lemma 4.8 of [11].

3. This follows from Theorem 2.8 of [11]. □

**Lemma 13.** *Let  $\mathcal{S}$  be a finite set of Horn (dual Horn) relations. If  $\mathcal{S}$  contains at least one relation that is not safely componentwise IHSB $-$  (not safely componentwise IHSB $+$ ),  $\text{CONN}_C(\mathcal{S})$  is coNP-complete.*

*Proof.* For sets of Horn relations that contain at least one relation that is not safely componentwise IHSB $-$ , the coNP-hardness follows from Lemmas 18 and 19 below. The case of sets of dual Horn relations that contain at least one relation that is not safely componentwise IHSB $+$  is analogous. Theorem 2.8 of [11] shows that  $\text{CONN}_C(\mathcal{S})$  is in coNP. □

To show the coNP-hardness of  $M$ , we first develop a criterion for the disconnectivity of  $G(\phi)$  for Horn formulas  $\phi$  (Lemma 17). Therefor, we introduce the following terms for Horn formulas; they are also used in the proof of Lemma 19.

**Definition 14.** *Clauses with only one literal are called unit clauses (positive if the literal is positive, negative otherwise). Clauses with only negative literals are restraints, and the sets of variables occurring in restraints are restraint sets. Clauses having one positive and one or more negative literals are implications. Implications with two or more negative literals are multi-implications.*

*We say a variable  $x$  has a branch to a variable  $y$ , if there is an implication  $y \vee \bar{x} \vee \bar{x}_1 \vee \dots \vee \bar{x}_k$ ,  $k \geq 0$ . A variable  $x$  is implied by a set of variables  $U$ , if setting all variables from  $U \setminus \{x\}$  to 1 forces  $x$  to be 1 in any satisfying assignment. A set of variables  $U$  is self-implicating if every  $x \in U$  is implied by  $U$ .  $U$  is maximal self-implicating, if it contains all variables implied by  $U$ .*

*Remark 15.* A Horn formula can be represented by a directed hypergraph with hyperedges of head-size one as follows: For every variable, there is a node, for every implication  $y \vee \bar{x}_1 \vee \dots \vee \bar{x}_k$ , there is a directed hyperedge from  $x_1, \dots, x_k$  to  $y$ , for every restraint  $\bar{x}_1 \vee \dots \vee \bar{x}_k$ , there is a directed hyperedge from  $x_1, \dots, x_k$  to a special node labeled “false”, and for every positive unit clause  $x$ , there is a directed hyperedge from a special node labeled “true” to  $x$ . For simplicity, we omit the “false” and “true” nodes and let the corresponding hyperedges end, resp. begin, in the void.

We draw the directed hyperedges as joining lines, e.g.,  $x \vee \bar{y} \vee \bar{z} =$  . See the following figures for more examples.

**Lemma 16.** *The solution graph  $G(\phi)$  of a Horn formula  $\phi$  without positive unit clauses is disconnected iff  $\phi$  has a locally minimal nonzero solution.*

*Proof.* This follows from Lemma 4.5 of [11] since the all-zero vector is a solution of every Horn formula without positive unit clauses, and Horn formulas are safely OR-free by Lemma 4.2 of [11].  $\square$

**Lemma 17.** *The solution graph  $G(\phi)$  of a Horn formula  $\phi$  without positive unit clauses is disconnected iff  $\phi$  has a non-empty maximal self-implicating set  $U$  containing no restraint set.*

*Proof.* If  $G(\phi)$  is disconnected,  $\phi$  has a locally minimal nonzero solution  $\mathbf{s}$  by Lemma 16. Let  $U$  be the set of variables assigned 1 in  $\mathbf{s}$ . Since  $\mathbf{s}$  is locally minimal, setting any variable  $x$  from  $U$  to 0 makes  $\phi$  false, so  $x$  must be implied by  $U$ , i.e.,  $x$  must appear as the positive literal in an implication clause with all variables from  $U$ . Thus  $U$  is self-implicating. Also,  $U$  must be non-empty, maximal self-implicating and can contain no restraint set since  $\mathbf{s}$  is a solution.

Conversely, if  $\phi$  has a non-empty maximal self-implicating set  $U$  containing no restraint set, the vector  $\mathbf{s}$  with all variables from  $U$  assigned 1, and all others 0, is a locally minimal nonzero solution: The implications  $\bar{y}_1 \vee \dots \vee \bar{y}_k \vee x$  with some  $y_i \notin U$  are satisfied with  $y_i = 0$ , and for the ones with all  $y_i \in U$ , also  $x \in U$  since  $U$  is maximal, so these are satisfied with  $x = 1$ . All restraints are satisfied since  $U$  contains no restraint set.  $\mathbf{s}$  is locally minimal since every variable assigned 1 is implied by  $U$ .  $\square$

**Lemma 18.**  $\text{CONN}_C(\{(x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)\})$  is CONP-hard.

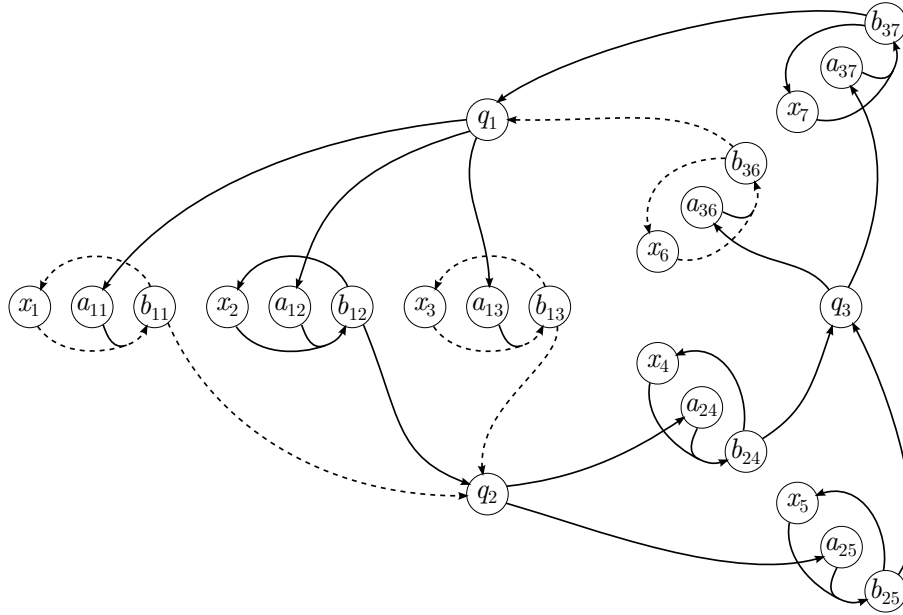
*Proof.* We reduce the no-constants satisfiability problem  $\text{SAT}(\{X, Y\})$  with  $X = x \vee y \vee z$  and  $Y = \bar{x} \vee \bar{y}$  to the complement of  $\text{CONN}_C(\{M\})$ , where  $M = (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)$ .  $\text{SAT}(\{X, Y\})$  is NP-hard by Schaefer's dichotomy theorem (Theorem 2.1 in [18]) since  $X$  is not 0-valid, not bijunctive, not Horn and not affine, while  $Y$  is not 1-valid and not dual Horn.

Let  $\psi$  be any CNF( $\{X, Y\}$ )-formula. If  $\psi$  only contains  $Y$ -constraints, it is trivially satisfiable, so assume it contains at least one  $X$ -constraint. We construct a CNF $_C(\{M\})$  formula  $\phi$  s.t. the solution graph  $G(\phi)$  is disconnected iff  $\psi$  is satisfiable (for an example see Figure 5). First note that we can use the relations  $\bar{x} \vee \bar{y} = M(0, x, y)$  and  $\bar{x} \vee y = M(x, 0, y)$ .

For every variable  $x_i$  of  $\psi$  ( $i = 1, \dots, n$ ), there is the same variable  $x_i$  in  $\phi$ . For every  $Y$ -constraint  $\bar{x}_i \vee \bar{x}_j$  of  $\psi$ , there is the clause  $\bar{x}_i \vee \bar{x}_j$  in  $\phi$  also. For every  $X$ -constraint  $c_p = x_{i_p} \vee x_{j_p} \vee x_{k_p}$  ( $p = 1, \dots, m$ ) of  $\psi$  there is an additional variable  $q_p$  in  $\phi$ , and for every  $x_l \in \{x_{i_p}, x_{j_p}, x_{k_p}\}$  appearing in  $c_p$ , there are two more additional variables  $a_{pl}$  and  $b_{pl}$  in  $\phi$ . Now for every  $c_p$ , for each  $l \in \{i_p, j_p, k_p\}$  the constraints  $\bar{q}_p \vee a_{pl}$ ,  $(\bar{x}_l \vee \bar{a}_{pl} \vee b_{pl}) \wedge (\bar{b}_{pl} \vee x_l)$  and  $\bar{b}_{pl} \vee q_r$  are added to  $\phi$ , where  $r = p + 1$  if  $p < m$ , and  $r = 1$  if  $p = m$ .


If  $\psi$  is satisfiable, there is an assignment  $\mathbf{s}$  to the variables  $x_i$  s.t. for every  $X$ -constraint  $c_p$  there is at least one  $x_l \in \{x_{i_p}, x_{j_p}, x_{k_p}\}$  assigned 1, and for no  $Y$ -constraint  $\bar{x}_i \vee \bar{x}_j$ , both  $x_i$  and  $x_j$  are assigned 1. We extend  $\mathbf{s}$  to a locally minimal nonzero satisfying assignment  $\mathbf{s}'$  for  $\phi$  ( $\mathbf{s}'$  must be nonzero since  $\psi$  contains at least one  $X$ -constraint); then  $G(\phi)$  is disconnected by Lemma 16: Let all  $q_p = 1$ ,  $a_{pl} = 1$ , and all  $b_{pl} = x_l$  in  $\mathbf{s}'$ . It is easy to check that all clauses of  $\phi$  are satisfied, and that all variables assigned 1 appear as the positive literal in an implication clause with all its variables assigned 1, so that  $\mathbf{s}'$  is locally minimal.

Conversely, if  $G(\phi)$  is disconnected,  $\phi$  has a maximal self-implicating set  $U$  containing no restraint set by Lemma 17. By starting with any set containing one of  $\phi$ 's variables and



**Figure 4.** An example for the proof of Lemma 18, illustrating the idea. Depicted here is the hypergraph representation (see Remark 15) of  $\phi$  for  $\psi = (x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5) \wedge (x_6 \vee x_7)$ , as constructed in the proof.

A maximal self-implicating set  $U$  of  $\phi$  corresponds to a big “circulatory” with every variable assigned 1 that passes through each  $q_p$  and at least one gadget  $(\bar{x}_l \vee \bar{a}_{pl} \vee b_{pl}) \wedge (\bar{b}_{pl} \vee x_l) =$



for each  $p$ ; these gadgets act as “valves”:  $b_{pl}$  may only be assigned 1 if  $x_l$  may be assigned 1.

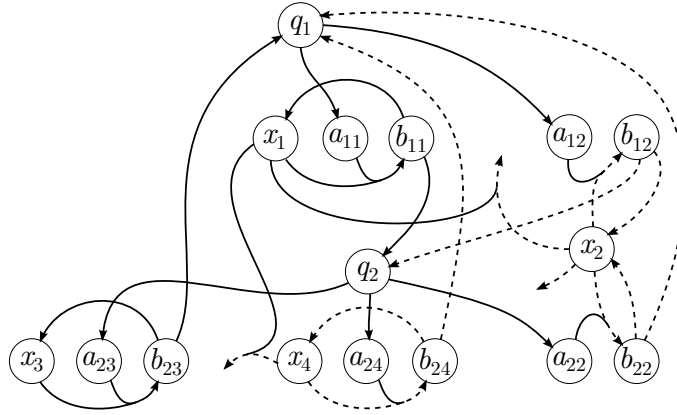
For example,  $U$  could consist of the variables with the outgoing edges drawn solid; then each such variable would be implied by  $U$ .  $\psi$  would become unsatisfiable, and  $G(\phi)$  connected, e.g. if the clauses  $\bar{x}_4 \vee \bar{x}_6$ ,  $\bar{x}_4 \vee \bar{x}_7, \bar{x}_5 \vee \bar{x}_6$  and  $\bar{x}_5 \vee \bar{x}_7$  were added to  $\psi$ .

extending it until it is maximal self-implicating, one finds that  $U$  must contain all  $q_p$ , all  $a_{pl}$ , and for every  $p$  for at least one  $l \in \{i_p, j_p, k_p\}$  both  $b_{pl}$  and  $x_l$ :

E.g., suppose  $U$  contains some  $x_l$ . Then it must also contain  $b_{pl}$  for some  $p$ , hence also  $a_{pl}$ , and then  $q_p$  as well. This means that  $U$  must contain all  $a_{pl'}$  and some  $b_{rl''}$ , where  $r = p - 1$  if  $p > 1$ , and  $r = m$  if  $p = 1$ , and  $l'' \in \{i_r, j_r, k_r\}$ . Therefore  $U$  must contain  $a_{rl''}$  and  $x_{l''}$ , and the implication starts over. For starting with some  $a_{pl}$ , some  $b_{pl}$ , or some  $q_p$ , the reasoning is similar.

Then the assignment with all  $x_i \in U$  assigned 1 and all other  $x_i$  assigned 0 satisfies  $\psi$  since  $U$  contains no restraint set.  $\square$

**Lemma 19.** *The relation  $M = (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)$  is expressible as a  $\text{CNF}_{\mathbb{C}}(\{R\})$ -formula for every Horn relation  $R$  that is not safely componentwise IHSB-.*



**Figure 5.** A more complex example as in Figure 4, with a variable of  $\psi$  appearing twice in an  $X$ -constraint:  $\phi$  for  $\psi = (x_1 \vee x_2) \wedge (x_3 \vee x_4 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_4) \wedge (\bar{x}_2)$ .

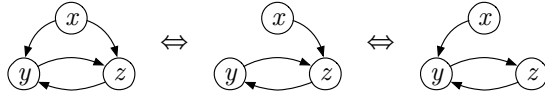
$\psi$  is satisfiable with the unique solution  $x_1 = x_3 = 1$  and  $x_2 = x_4 = 0$ , and  $G(\phi)$  is disconnected (with exactly two components, since there is exactly one maximal self-implicating set containing no restraint set, consisting of the variables with the outgoing edges drawn solid).

*Proof.* We show that  $M = (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z) = \textcircled{x} \textcircled{y} \textcircled{z}$ ,  $L = (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{y} \vee z) = \textcircled{x} \textcircled{y} \textcircled{z}$ , or  $K = x \vee \bar{y} \vee \bar{z} = \textcircled{x} \textcircled{y} \textcircled{z}$  is expressible from  $R$  by substituting constants and identifying variables. We can then express  $M$  from  $K$  or  $L$  as

$$M(x, y, z) \equiv K(x, y, z) \wedge K(z, x, x) \equiv L(x, y, z) \wedge L(z, x, x).$$

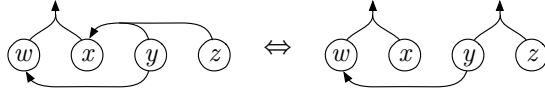
The following steps generate  $K$ ,  $L$ , or  $M$  from  $R$ .

1. Obtain a not componentwise IHSB- relation  $R^*$  from  $R$  by identification of variables.
2. Assign all variables that can take only one value in  $R^*$  that value to obtain a relation  $R'$ . It is clear that  $R'$  still is Horn and not componentwise IHSB-. Now a CNF-formula representation  $\phi'$  of  $R'$  contains no unit clauses, and identification of variables cannot make  $\phi'$  unsatisfiable since there are no clauses with only positive literals (the all-zero vector is always a solution).
3. Simplify the formula such that the following conditions hold; it is easy to check that the listed operations do not change the value of the formula and have the stated effects, and that the conditions are retained throughout the subsequent steps:
  - (a) *No redundant implications:* Remove a clause  $c = x \vee \bar{y}_1 \vee \dots \vee \bar{y}_k$  ( $k \geq 1$ ) if  $x$  is already implied by  $\{y_1, \dots, y_k\}$  via other clauses. If there are many possibilities, choose one. Repeat as long as there is such a clause in  $\phi'$ .  
E.g.,  $c$  would be removed if there were a clause  $x \vee \bar{y}_{i_1} \vee \dots \vee \bar{y}_{i_l}$  with  $\{y_{i_1}, \dots, y_{i_l}\} \subset \{y_1, \dots, y_k\}$ , or clauses  $x \vee \bar{q}$  and  $q \vee \bar{y}_1 \vee \dots \vee \bar{y}_k$ . If  $\phi'$  were  $(\bar{x} \vee y) \wedge (\bar{x} \vee z) \wedge (\bar{z} \vee y) \wedge (\bar{y} \vee z)$ , then  $\bar{x} \vee y$  or  $\bar{x} \vee z$  would be removed:



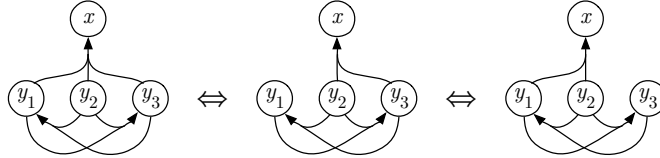
Now there are no implications where the positive literal is already implied by the negative literals via other clauses.

- (b) *No impossible implications:* Replace every clause  $x \vee \bar{y}_1 \vee \dots \vee \bar{y}_k$  by  $\bar{y}_1 \vee \dots \vee \bar{y}_k$  if there is a restraint where all variables are from, or implied by,  $\{y_1, \dots, y_k\}$ . E.g., if there is a clause  $\bar{r}_1 \vee \dots \vee \bar{r}_l$  with  $\{r_1, \dots, r_l\} \subseteq \{x, y_1, \dots, y_k\}$ , or clauses  $q_1 \vee \bar{r}_1 \vee \dots \vee \bar{r}_l$ ,  $q_2 \vee \bar{r}_1 \vee \dots \vee \bar{r}_l$  and  $\bar{q}_1 \vee \bar{q}_2$ . If  $\phi'$  were  $(x \vee \bar{y} \vee \bar{z}) \wedge (w \vee \bar{y}) \wedge (\bar{w} \vee \bar{x})$ ,  $x \vee \bar{y} \vee \bar{z}$  would be replaced by  $\bar{y} \vee \bar{z}$ :



Now there are no restraints with only variables implied by the variables of an implication clause.

- (c) *No redundant branches:* Remove the literal  $\bar{y}_i \in \{\bar{y}_1, \dots, \bar{y}_k\}$  from a multi-implication  $x \vee \bar{y}_1 \vee \dots \vee \bar{y}_k$  if  $y_i$  is implied by a set  $\{y_{i_1}, \dots, y_{i_l}\} \subset \{y_1, \dots, y_k\}$ ,  $y_i \notin \{y_{i_1}, \dots, y_{i_l}\}$ . If there are many possibilities (e.g. if there are clauses  $y_1 \vee \bar{y}_2 \vee \bar{y}_3$  and  $\bar{y}_1 \vee \bar{y}_2 \vee y_3$ ), choose one:



Repeat this for every multi-implication as long as there is such a literal.

Now there are no implications among the negative literals of implication clauses.

Since  $\phi'$  is not IHSB-, it contains a multi-implication. We show that since  $\phi'$  is also not componentwise IHSB-, there must be a multi-implication  $c$  s.t.  $\text{Var}(c)$  is not self-implicating:

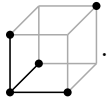
If the set of variables of an implication  $s_1 = x_1 \vee \bar{x}_2 \vee \dots \vee \bar{x}_k$  is self-implicating, we can add the clauses

$$s_2 = x_2 \vee \bar{x}_3 \vee \dots \vee \bar{x}_k \vee \bar{x}_1, s_3 = x_3 \vee \bar{x}_4 \vee \dots \vee \bar{x}_k \vee \bar{x}_1 \vee \bar{x}_2, \dots, s_k = x_k \vee \bar{x}_1 \vee \dots \vee \bar{x}_{k-1}$$

to  $\phi'$  without changing its value. Now it is easy to see that  $s_1 \wedge \dots \wedge s_k \equiv d \vee e$ , with IHSB- expressions

$$d = (\bar{x}_2 \vee \dots \vee \bar{x}_k) \wedge (\bar{x}_3 \vee \dots \vee \bar{x}_k \vee \bar{x}_1) \wedge \dots \wedge (\bar{x}_1 \vee \dots \vee \bar{x}_{k-1}), e = x_1 \wedge \dots \wedge x_k,$$

and that the solutions of  $d$  are not connected to that of  $e$  in the solution graph  $G(d \vee e)$  for  $k \geq 3$ . E.g., for  $k = 3$ ,  $(x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee y \vee \bar{z}) \wedge (\bar{x} \vee \bar{y} \vee z) \equiv ((\bar{y} \vee \bar{z}) \wedge (\bar{z} \vee \bar{x}) \wedge (\bar{x} \vee \bar{y})) \vee (x \wedge y \wedge z) =$



Let  $R' = c \wedge t_1 \wedge \dots \wedge t_l$ , where  $t_1, \dots, t_l$  are the multi-implications, and  $c = c_1 \wedge \dots \wedge c_k$  all other clauses. Now if for every  $t_i$ ,  $\text{Var}(t_i)$  were self-implicating,  $R'$  could be written as

$$R' = c \wedge (d_1 \vee e_1) \wedge \dots \wedge (d_l \vee e_l)$$



with IHSB– expressions  $d_1, \dots, d_l$  and  $e_1, \dots, e_l$ , and the solutions of no  $d_i$  connected to that of  $e_i$  in the solution graph  $G(d_i \vee e_i)$ . But then the expansion of the conjunction,

$$R' = (c \wedge d_1 \wedge \dots \wedge d_l) \vee (c \wedge d_1 \wedge \dots \wedge d_{l-1} \wedge e_l) \vee \dots \vee (c \wedge e_1 \wedge \dots \wedge e_l),$$

would yield a partition of  $R'$  into pairwise disconnected (possibly empty) IHSB– parts (for each pair of disjuncts, there is at least one  $i \in \{1, \dots, l\}$  with  $d_i$  appearing in one disjunct and  $e_i$  in the other, and thus the projections to the variables appearing in  $d_i$  and  $e_i$  are already disconnected). Since every component of an IHSB– relation is itself IHSB– by Lemma 4.1 of [11],  $R'$  would then be componentwise IHSB–.

Thus  $R'$  must contain a multi-implication  $c$  s.t.  $\text{Var}(c)$  is not self-implicating, and then this  $c$  has a literal  $\bar{y}$  s.t.  $y$  is not implied by  $\text{Var}(c)$ , and the following step is possible:

4. *Choose some multi-implication  $c = x \vee \bar{y} \vee \bar{z}_1 \vee \dots \vee \bar{z}_k$  ( $k \geq 1$ ) such that  $\{x, y, z_1, \dots, z_k\}$  is not self-implicating, with  $y$  being a variable not implied by  $\{x, z_1, \dots, z_k\}$ . Then there is no clause  $x \vee \bar{a}_1 \vee \dots \vee \bar{a}_l$  with  $\{a_1, \dots, a_l\} \subseteq \{x, z_1, \dots, z_k\}$ . Also, by (3a), there is no clause  $x \vee \bar{a}_1 \vee \dots \vee \bar{a}_l$  with  $\{a_1, \dots, a_l\} \subsetneq \{y, z_1, \dots, z_k\}$ , and by (3c) no clause  $a_0 \vee \bar{a}_1 \vee \dots \vee \bar{a}_l$  with  $\{a_0, a_1, \dots, a_l\} \subseteq \{y, z_1, \dots, z_k\}$ .*

Next we trim  $c$  to size 3:

5. *Identify  $z_1, \dots, z_k$ , call the resulting variable  $z$ . This produces the clause  $x \vee \bar{y} \vee \bar{z}$  from  $c$ . It is easy to see that consequentially there are no clauses  $x \vee \bar{y}$ ,  $x \vee \bar{z}$ , no  $y \vee \bar{x}$ ,  $y \vee \bar{z}$ ,  $y \vee \bar{x} \vee \bar{z}$ , and no  $z \vee \bar{y}$ . Thus other clauses only involving  $x, y, z$  can be only from  $\{z \vee \bar{x}, z \vee \bar{x} \vee \bar{y}\}$ . Also, no unit clauses can emerge, by (3b).*

Now we eliminate all restraints:

6. *Set all remaining variables not implied by  $\{x, y, z\}$  to 0. This eliminates all restraints because they must contain at least one variable not implied by  $\{x, y, z\}$ , and produces no new restraints since any implication with the positive literal not implied by  $\{x, y, z\}$  must also have some negative literal not implied by  $\{x, y, z\}$ , and thus vanishes. Also, no unit clauses can emerge.*

All remaining variables are now implied by  $\{x, y, z\}$ , and thus can have no branch to  $y$ . We next handle the variables that also imply variables from  $\{x, y, z\}$ . We can't set them to constants because this would eventually produce unit clauses or restraints for the variables  $x, y, z$ , so we identify them with  $x, y$  or  $z$ . We also must be careful not to produce unwanted implications among  $\{x, y, z\}$ .

7. *Identify all remaining variables  $x_i \notin \{y, z\}$  having a branch to  $x$  with  $x$ , repeat this as long as there are such  $x_i$ . These  $x_i$  cannot have been implied by  $\{y, z\}$  due to (3a); thus there cannot emerge an implication  $x \vee \bar{y}$ ,  $x \vee \bar{z}$  (or  $x \vee \bar{y} \vee \bar{z}$ ). Also, the set of these  $x_i$  cannot have implied  $y$  because of (6) since  $y$  is not implied by  $\{x, z\}$ , and thus there neither can emerge an implication  $\bar{x} \vee y$ .*

8. Identify all remaining variables  $x_i \notin \{x, z\}$  having a branch to  $z$  with  $z$ , repeat this as long as there are such  $x_i$ .

These  $x_i$  cannot have been implied by  $\{y\}$  due to (3c); thus there cannot emerge an implication  $z \vee \bar{y}$ , only  $z \vee \bar{x}$  or  $z \vee \bar{x} \vee \bar{y}$ . Analogously to (7), there neither can emerge an implication  $y \vee \bar{z}$  or  $x \vee \bar{z}$ .

All remaining variables other than  $x, y, z$  now have no branch to any of  $x, y, z$ , and so we can set them to 1 (since they also do not occur in restraints).

9. Set all remaining variables other than  $x, y, z$  to 1.

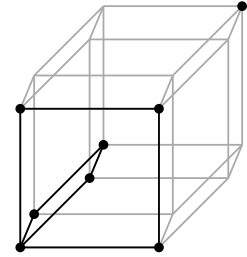
We are now left with  $K, L$ , or  $M$ . □

## 5. Constraint-Projection Separating Sets of Relations

In this supplemental section we reveal a common property of all CPSS sets  $\mathcal{S}$  of relations and derive a simple algorithm for  $\text{CONN}_{\mathcal{C}}(\mathcal{S})$ .

**Definition 20.** A set  $\mathcal{S}$  of logical relations is constraint-projection separating, if for every  $\text{CNF}_{\mathcal{C}}(\mathcal{S})$ -formula  $\phi$  whose solution graph  $G(\phi)$  is disconnected, there exists a constraint  $C_i$  s.t.  $G(\phi_i)$  is disconnected, where  $\phi_i$  is the projection of  $\phi$  to  $\text{Var}(C_i)$ .

For example,  $\{x \vee \bar{y}\}$  is projection-separating (for the proof see below); e.g., for  $(x \vee \bar{y}) \wedge (y \vee \bar{z}) \wedge (z \vee \bar{x})$ , the projections to  $\{x, y\}$ ,  $\{y, z\}$  and  $\{z, x\}$  are all disconnected. In contrast,  $\{x \vee \bar{y} \vee \bar{z}\}$  is not projection-separating: E.g.,  $(x \vee \bar{y} \vee \bar{z}) \wedge (y \vee \bar{z} \vee \bar{w}) \wedge (z \vee \bar{w} \vee \bar{x}) \wedge (w \vee \bar{x} \vee \bar{y})$  (see the graph on the right) is disconnected, but the projection to any three variables is connected.



In Lemma 24 we show that CPSS sets of relations, as defined in Definition 9, are indeed constraint-projection separating, so that the following algorithm works.

**Theorem 21.** Let  $\mathcal{S}$  be a CPSS set of relations, and  $\phi$  a  $\text{CNF}_{\mathcal{C}}(\mathcal{S})$ -formula. Then, the following polynomial-time algorithm decides whether  $G(\phi)$  is connected:

For every constraint  $C_i$  of  $\phi$ , obtain the projection  $\phi_i$  of  $\phi$  to the variables  $\mathbf{x}^i$  occurring in  $C_i$  by checking for every assignment  $\mathbf{a}$  of  $\mathbf{x}^i$  whether  $\phi[\mathbf{x}^i/\mathbf{a}]$  is satisfiable. Then  $G(\phi)$  is connected iff for no  $\phi_i$ ,  $G(\phi_i)$  is disconnected ( $\phi[\mathbf{x}^i/\mathbf{a}]$  denotes the formula with the constants  $a_j$  substituted for the variables  $x_j^i$ ).

*Proof.* Every projection can be computed in polynomial time since  $\phi$  is Schaefer, and connectivity of every  $G(\phi_i)$  can be checked in constant time. If  $G(\phi)$  is disconnected, some  $G(\phi_i)$  is disconnected since  $\phi$  is constraint-projection separating by Lemma 24 below. If some  $G(\phi_i)$  is disconnected, it is obvious that  $G(\phi)$  cannot be connected. □

**Lemma 22.** Let  $\mathcal{S}$  be a set of IHSB– (IHSB+) relations and  $\phi$  a  $\text{CNF}_{\mathcal{C}}(\mathcal{S})$ -formula. Then for any two components of  $G(\phi)$ , there is some constraint  $C_i$  of  $\phi$  s.t. their images in the projection  $\phi_i$  of  $\phi$  to  $\text{Var}(C_i)$  are disconnected in  $G(\phi_i)$ .

*Proof.* We prove the IHSB− case, the IHSB+ case is analogous. Consider any two components  $A$  and  $B$  of  $\phi$ . Since every IHSB− relation is OR-free, there is a locally minimal solution  $\mathbf{a}$  in  $A$  and a locally minimal solution  $\mathbf{b}$  in  $B$  by Lemma 4.5 of [11]. Let  $U$  and  $V$  be the sets of variables that are assigned 1 in  $\mathbf{a}$  and  $\mathbf{b}$ , resp. At least one of the sets  $U' = U \setminus V$  or  $V' = V \setminus U$  is not empty, assume it is  $U'$ . Then for every  $x_1 \in U'$  there must be a clause  $x_1 \vee \bar{x}_2$  with  $x_2 \in U$  since  $\mathbf{a}$  is locally minimal, and also  $x_2$  must be from  $U'$ , else  $\mathbf{b}$  would not be satisfying.

But then for  $x_2$  there must be also some variable  $x_3 \in U'$  and a clause  $x_2 \vee \bar{x}_3$ , and we can add the clause  $x_1 \vee \bar{x}_3$  to  $\phi$  without changing its value. Continuing this way, we will find a cycle, i.e. a clause  $x_i \vee \bar{x}_{i+1}$  with  $x_{i+1} = x_j$ ,  $j < i$ . But then we already have  $x_j \vee \bar{x}_i$  added, thus  $(s_i, s_j) \in \{(0, 0), (1, 1)\}$  for any solution  $\mathbf{s}$  of  $\phi$ , and there must be some constraint  $C_i$  with both  $x_i$  and  $x_j$  occurring in it (the  $C_i$  in which the original  $x_i \vee \bar{x}_j$  appeared), and thus the projections of  $A$  and  $B$  to  $\text{Var}(C_i)$  are disconnected in  $G(\phi_i)$ .  $\square$

**Lemma 23.** *Let  $\mathcal{S}$  be a set of bijunctive relations and  $\phi$  a  $\text{CNF}_C(\mathcal{S})$ -formula. Then for any two components of  $G(\phi)$ , there is some constraint  $C_i$  of  $\phi$  s.t. their images in the projection  $\phi_i$  of  $\phi$  to  $\text{Var}(C_i)$  are disconnected in  $G(\phi_i)$ .*

*Proof.* The proof is similar to the last one. Consider any two components  $A$  and  $B$  of  $\phi$  and two solutions  $\mathbf{a}$  in  $A$  and  $\mathbf{b}$  in  $B$  that are at minimum Hamming distance. Let  $L$  be the set of literals that are assigned 1 in  $\mathbf{a}$ , but assigned 0 in  $\mathbf{b}$ . Then for every  $l_1 \in L$  that is assigned 1 in  $\mathbf{a}$ , there must be a clause equivalent to  $l_1 \vee \bar{l}_2$  in  $\phi$  s.t.  $l_2$  is also assigned 1 in  $\mathbf{a}$ , else the variable corresponding to  $l_1$  could be flipped in  $\mathbf{a}$ , and the resulting vector would be closer to  $\mathbf{b}$ , contradicting our choice of  $\mathbf{a}$  and  $\mathbf{b}$ . Also,  $l_2$  must be assigned 0 in  $\mathbf{b}$ , i.e.  $l_2 \in L$ , else  $\mathbf{b}$  would not be satisfying.

But then for  $l_2$  there must be also some literal  $l_3 \in L$  that is assigned 1 in  $\mathbf{a}$  and a clause equivalent to  $l_2 \vee \bar{l}_3$  in  $\phi$ , and we can add the clause  $l_1 \vee \bar{l}_3$  to  $\phi$  without changing its value. Continuing this way, we will find a cycle, i.e. a clause equivalent to  $l_n \vee \bar{l}_{n+1}$  with  $l_{n+1} = l_m$ ,  $m < n$ . But then we already have  $l_m \vee \bar{l}_n$  added, thus if  $x_i$  and  $x_j$  are the variables corresponding to  $l_n$  resp.  $l_m$ , then  $(s_i, s_j) \in \{(0, 1), (1, 0)\}$  (if  $l_n$  and  $l_m$  were both positive or both negative), or  $(s_i, s_j) \in \{(0, 0), (1, 1)\}$  (otherwise), for any solution  $\mathbf{s}$  of  $\phi$ . Also, there must be some constraint  $C_i$  with both  $x_i$  and  $x_j$  occurring in it (the constraint in which the clause equivalent to  $l_n \vee \bar{l}_m$  appeared), and thus the projections of  $A$  and  $B$  to  $\text{Var}(C_i)$  are disconnected in  $G(\phi_i)$ .  $\square$

**Lemma 24.** *Every set  $\mathcal{S}$  of safely componentwise bijunctive (safely componentwise IHSB−, safely componentwise IHSB+, affine) relations is constraint-projection separating.*

*Proof.* The affine case follows from the safely componentwise bijunctive case since every affine relation is safely componentwise bijunctive by Lemma 4.2 of [11].

If the relation corresponding to some  $C_i$  is disconnected, and there is more than one component of this relation for which  $\phi$  has solutions with the variables of  $C_i$  assigned values in that component, the projection of  $\phi$  to  $\text{Var}(C_i)$  must be disconnected in  $G(\phi_i)$ .

So assume that for every constraint  $C_i$ ,  $\phi$  only has solutions in which the variables of  $C_i$  are assigned values in one component  $P_i$  of the relation corresponding to  $C_i$ . Then we can replace every  $C_i$  with  $P_i$  to obtain an equivalent formula  $\phi'$ . Since  $\mathcal{S}$  is safely componentwise bijunctive (safely componentwise IHSB−, safely componentwise IHSB+),

each  $P_i$  is bijunctive (IHSB−, IHSB−), and thus so is  $\phi'$ , and the statement follows from Lemmas 22 and 23.  $\square$

*Remark 25.* The Lemmas 22 and 23 cannot be generalized to safely componentwise bijunctive or safely componentwise IHSB− relations: For sets  $\mathcal{S}$  of safely componentwise bijunctive (safely componentwise IHSB−) relations that are not bijunctive (IHSB−), there are  $\text{CNF}_C(\mathcal{S})$ -formulas with pairs of components that are not disconnected in the projection to any constraint.

E.g., for the safely componentwise bijunctive relation  $R = ((x \vee \bar{y}) \wedge \bar{z}) \vee (\bar{x} \wedge y \wedge z)$ , the  $\text{CNF}_C(\{R\})$ -formula  $F(x, y, z, w) = R(x, y, z) \wedge R(y, x, w)$  has the four pairwise disconnected solutions  $a=0000$ ,  $b=1100$ ,  $c=0110$ , and  $d=1001$ , but  $a$  is connected to  $b$  in the projection to  $\{x, y, z\}$  as well as in the one to  $\{x, y, w\}$ .

Finally, we show that Schaefer sets of relations that are not CPSS are not constraint-projection separating. Lemma 24 shows that there are non-Schaefer sets that are constraint-projection separating. It is open whether there are other such sets not mentioned in Lemma 24.

**Lemma 26.** *If a set of relations  $\mathcal{S}$  is Schaefer but not CPSS, there is a  $\text{CNF}_C(\mathcal{S})$ -formula  $\phi$  that is not constraint-projection separating.*

*Proof.* Since  $\mathcal{S}$  is Schaefer but not CPSS, it must contain some relation that is Horn but not safely componentwise IHSB−, or dual Horn but not safely componentwise IHSB+. Assume the first case, the second one is analogous. Then by Lemma 19, we can express  $M = (x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)$  as a  $\text{CNF}_C(\mathcal{S})$ -formula. Consider the  $\text{CNF}_C(\mathcal{S})$ -formula

$$\begin{aligned} T(u, v, w, x, y, z) &= M(u, v, w) \wedge M(x, y, z) \wedge M(w, w, y) \wedge M(z, z, v) \\ &\equiv ((u \vee \bar{v} \vee \bar{w}) \wedge (\bar{u} \vee w)) \wedge ((x \vee \bar{y} \vee \bar{z}) \wedge (\bar{x} \vee z)) \wedge (y \vee \bar{w}) \wedge (v \vee \bar{z}). \end{aligned}$$

Now  $G(T)$  is disconnected by Lemma 17 since  $\{u, v, w, x, y, z\}$  is maximal self-implicating, but neither the projection  $\exists x \exists y \exists z T \equiv M(u, v, w)$  to the variables of the first constraint in the  $\text{CNF}(\{M\})$ -representation of  $T$ , nor the projection  $\exists u \exists v \exists x \exists z T \equiv y \vee \bar{w}$  to the variables of the third one is disconnected. The second and fourth constraints are symmetric to the first and third ones. Since in the  $\text{CNF}_C(\mathcal{S})$ -representation of  $T$  every conjunct  $M(r, s, t)$  of  $T$  ( $r, s, t \in \{u, v, w, x, y, z\}$ ) is a  $\text{CNF}_C(\mathcal{S})$ -formula  $\bigwedge_i R_i(\xi^i)$  with  $R_i \in \mathcal{S}$  and  $\xi_j^i \in \{0, 1, r, s, t\}$ , for every constraint  $C_i$  of  $T$ , the set  $\text{Var}(C_i)$  is a subset of  $\{u, v, w\}$ ,  $\{x, y, z\}$ ,  $\{y, w\}$  or  $\{v, z\}$ , and thus also for no  $C_i$  the projection to  $\text{Var}(C_i)$  is disconnected.  $\square$

## 6. Related and Future Work

Now that the connectivity and  $st$ -connectivity questions as well as the diameter seem to be settled for  $\text{CNF}_C(\mathcal{S})$ -formulas, a classification for  $\text{CNF}(\mathcal{S})$ -formulas without constants would be interesting, as Gopalan et al. already remarked [11]. Another variation are partially quantified formulas. We have obtained partial results for both cases [20], but a complete classification is still missing.

Besides CSPs in Schaefer's framework, there are other ways to look at Boolean satisfiability. Disjunctive normal forms with special connectivity properties were studied by Ekin et

al. already in 1997 for their “important role in problems appearing in various areas including in particular discrete optimization, machine learning, automated reasoning, etc.” [9].

Recently, we have investigated the connectivity for  $B$ -formulas and  $B$ -circuits in Post’s framework, inter alia in view of circuit-based SAT solvers, and could prove a distinct dichotomy: For  $B \subseteq M$ ,  $B \subseteq L$ , or  $B \subseteq S_0$ , both connectivity problems are in P and the diameter is linear, while in all other cases, the problems are PSPACE-complete and the diameter can be exponential [19]. Here, we also could obtain results for partially quantified formulas already.

There are yet more kinds of representations of Boolean relations, such as binary decision diagrams and or Boolean neural networks, and investigating the connectivity in these settings might be worthwhile as well.

Other connectivity-related problems already mentioned by Gopalan et al. are counting the number of components and approximating the diameter. Also, with regard to reconfiguration problems, one could try to find the shortest path between two solutions, or the optimal path according to some measure.

Furthermore, our definition of connectivity is not the only sensible one: One could regard two solutions connected whenever their Hamming distance is at most  $d$ , for any fixed  $d \geq 1$ ; this was already considered related to random satisfiability, see [1]. This generalization seems meaningful as well as challenging.

Finally, a most interesting subject are CSPs over larger domains; in 1993, Feder and Vardi conjectured a dichotomy for the satisfiability problem over arbitrary finite domains [10], and while the conjecture was proved for domains of size three in 2002 by Bulatov [5], it remains open to date for the general case. Close investigation of the solution space might lead to valuable insights here.

For  $k$ -colorability, which is a special case of the general CSP over a  $k$ -element set, the connectivity problems and the diameter were already studied by Bonsma and Cereceda [4], and Cereceda, van den Heuvel, and Johnson [6]. They showed that for  $k = 3$  the diameter is at most quadratic in the number of vertices and the  $st$ -connectivity problem is in P, while for  $k \geq 4$ , the diameter can be exponential and  $st$ -connectivity is PSPACE-complete in general.

## Acknowledgments

I am grateful to Heribert Vollmer for pointing me to these interesting themes.

## References

- [1] Dimitris Achlioptas and Federico Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 130–139. ACM, 2006.
- [2] Elmar Böhler, Nadia Creignou, Steffen Reith, and Heribert Vollmer. Playing with boolean blocks, part i: Posts lattice with applications to complexity theory. In *SIGACT News*, 2003.

- [3] Elmar Böhler, Steffen Reith, Henning Schnoor, and Heribert Vollmer. Bases for boolean co-clones. *Information Processing Letters*, **96**(2):59–66, 2005.
- [4] Paul Bonsma and Luis Cereceda. Finding paths between graph colourings: Pspace-completeness and superpolynomial distances. *Theoretical Computer Science*, **410**(50):5215–5226, 2009.
- [5] Andrei A Bulatov. A dichotomy theorem for constraints on a three-element set. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 649–658. IEEE, 2002.
- [6] Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Finding paths between 3-colorings. *Journal of graph theory*, **67**(1):69–82, 2011.
- [7] Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity classifications of boolean constraint satisfaction problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.
- [8] Nadia Creignou, Phokion Kolaitis, and Bruno Zanuttini. Structure identification of boolean relations and plain bases for co-clones. *Journal of Computer and System Sciences*, **74**(7):1103–1115, 2008.
- [9] Oya Ekin, Peter L Hammer, and Alexander Kogan. On connected boolean functions. *Discrete Applied Mathematics*, **96**:337–362, 1999.
- [10] Tomás Feder and Moshe Y Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, **28**(1):57–104, 1998.
- [11] Parikshit Gopalan, Phokion G. Kolaitis, Elitza Maneva, and Christos H. Papadimitriou. The connectivity of boolean satisfiability: Computational and structural dichotomies. *SIAM J. Comput.*, **38**(6):2330–2355, March 2009.
- [12] Parikshit Gopalan, Phokion G. Kolaitis, Elitza N. Maneva, and Christos H. Papadimitriou. The connectivity of boolean satisfiability: Computational and structural dichotomies. ICALP’06, pages 346–357, 2006.
- [13] Takehiro Ito, Erik D. Demaine, Nicholas J. A. Harvey, Christos H. Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theor. Comput. Sci.*, **412**(12-14):1054–1065, March 2011.
- [14] Marcin Kamiński, Paul Medvedev, and Martin Milanič. Shortest paths between shortest paths and independent sets. In *Combinatorial Algorithms*, pages 56–67. Springer, 2011.
- [15] Kazuhisa Makino, Suguru Tamaki, and Masaki Yamamoto. On the boolean connectivity problem for horn relations. In *Proceedings of the 10th international conference on Theory and applications of satisfiability testing, SAT’07*, pages 187–200, 2007.
- [16] Elitza Maneva, Elchanan Mossel, and Martin J Wainwright. A new look at survey propagation and its generalizations. *Journal of the ACM (JACM)*, **54**(4):17, 2007.

- [17] Marc Mézard, Thierry Mora, and Riccardo Zecchina. Clustering of solutions in the random satisfiability problem. *Physical Review Letters*, **94**(19):197205, 2005.
- [18] Thomas J. Schaefer. The complexity of satisfiability problems. STOC '78, pages 216–226, 1978.
- [19] Konrad W. Schwerdtfeger. The connectivity of boolean satisfiability: Dichotomies for formulas and circuits. *ArXiv CoRR*, **abs/1312.6679**, 2013. Extended version of a paper appearing in proceedings of the 9th International Computer Science Symposium in Russia, CSR 2014.
- [20] Konrad W. Schwerdtfeger. The connectivity of boolean satisfiability: No-constants and quantified variants. *ArXiv CoRR*, **abs/1403.6165**, 2014.