Electronic Colloquium on Computational Complexity, Report No. 94 (2006)

The Connectivity of Boolean Satisfiability: Computational and Structural Dichotomies

Parikshit Gopalan*
Georgia Tech.
parik@cc.gatech.edu

Phokion G. Kolaitis†
IBM Almaden.
kolaitis@us.ibm.com

Elitza N. Maneva*
UC Berkeley.
elitza@cs.berkeley.edu

Christos H. Papadimitriou
UC Berkeley.
christos@cs.berkeley.edu

July 29, 2006

Abstract

Boolean satisfiability problems are an important benchmark for questions about complexity, algorithms, heuristics and threshold phenomena. Recent work on heuristics, and the satisfiability threshold has centered around the structure and connectivity of the solution space. Motivated by this work, we study structural and connectivity-related properties of the space of solutions of Boolean satisfiability problems and establish various dichotomies in Schaefer's framework.

On the structural side, we obtain dichotomies for the kinds of subgraphs of the hypercube that can be induced by the solutions of Boolean formulas, as well as for the diameter of the connected components of the solution space. On the computational side, we establish dichotomy theorems for the complexity of the connectivity and st-connectivity questions for the graph of solutions of Boolean formulas. Our results assert that the intractable side of the computational dichotomies is PSPACE-complete, while the tractable side - which includes but is not limited to all problems with polynomial time algorithms for satisfiability - is in P for the st-connectivity question, and in coNP for the connectivity question. The diameter of components can be exponential for the PSPACE-complete cases, whereas in all other cases it is linear; thus, small diameter and tractability of the connectivity problems are remarkably aligned. The crux of our results is an expressibility theorem showing that in the tractable cases, the subgraphs induced by the solution space posses certain good structural properties, whereas in the intractable cases, the subgraphs can be arbitrary.

*Work done in part while the authors were interns at IBM Almaden.
†On leave from UC Santa Cruz

ISSN 1433-8092

1 Introduction

In 1978, T.J. Schaefer [20] introduced a rich framework for expressing variants of Boolean satisfiability and proved a remarkable *dichotomy theorem*: the satisfiability problem is in P for certain classes of Boolean formulas, while it is NP-complete for all other classes in the framework. In a single stroke, this result pinpoints the computational complexity of all well-known variants of SAT, such as 3-SAT, HORN 3-SAT, NOT-ALL-EQUAL 3-SAT, and 1-IN-3 SAT. Schaefer's work paved the way for a series of investigations establishing dichotomies for several aspects of satisfiability, including optimization [6, 8, 14], counting [7], inverse satisfiability [13], minimal satisfiability [15], 3-valued satisfiability [5] and propositional abduction [9].

Our aim in this paper is to carry out a comprehensive exploration of a different aspect of Boolean satisfiability, namely, the *c*onnectivity properties of the space of solutions of Boolean formulas. The solutions (satisfying assignments) of a given n-variable Boolean formula φ induce a subgraph $G(\varphi)$ of the n-dimensional hypercube, which we call the solution graph. We believe that connectivity properties of such graphs merit study in their own right, as they shed light on the structure of the solution space of Boolean formulas. Furthermore, in recent years the structure of the solution graph for random instances has been the main consideration at the basis of both algorithms for and mathematical analysis of the satisfiability problem [2, 18, 17, 16]. It has been conjectured for 3-SAT [17] and proved for 8-SAT [19, 3], that the solution space fractures as one approaches the *critical region* from below. This apparently leads to performance deterioration of the standard satisfiability algorithms, such as WalkSAT [21] and DPLL [1]. It is also the main consideration behind the design of the survey propagation algorithm, which has far superior performance on random instances of satisfiability [17]. This body of work has served as a motivation to us for pursuing the investigation reported here. While there has been an intensive study of the structure of the solution space of Boolean satisfiability problems for random instances, our work seems to be the first to explore this issue from a worst-case viewpoint.

Our Results. Our work addresses the question: *when does the solution graph of a Boolean formula have nice structure?* To answer this question, one must clarify what is meant by *nice structure*. One can define it in terms of graph theoretic properties of the solution graph. We can ask what kinds of graphs are possible as solution graphs of a Boolean formula. One can focus on specific structural properties such as diameter of each component. Alternatively, once can view the Boolean formula as an implicit description of the solution graph, and study the computational complexity of algorithmic tasks such as finding if the graph is connected. Surprisingly, we show that many of these properties, both structural and algorithmic, are remarkably aligned and result in the same dichotomies.

We identify two broad classes of Boolean relations[1] with respect to the structure of the solution graphs of Boolean formulas built using these relations, which we call tight and non-tight relations. The solution graphs of formulas built from tight relations are characterized by certain structural properties. On the other hand we find *non-tight* sets of relations; formulas built from such sets of relations can express any solution graph. The boundary between these two classes differs from the boundary in Schaefer's dichotomy. Schaefer showed that the satisfiability problem is solvable in polynomial time precisely for formulas built from Boolean relations all of which are bijunctive, or all of which are Horn, or all of which are dual Horn, or all of which are affine. The class of *tight* relations properly contains the classes of bijunctive, Horn, dual Horn, and affine relations.

The main step in the proof of Schaefer's dichotomy theorem is a result of independent interest known as Schaefer's expressibility theorem. The crux of our results is a different expressibility theorem which we call the *Faithful Expressibility Theorem*. At a high level, this theorem asserts that given any Boolean relation with a solution graph G, we can construct a formula using any non-tight set of relations, such that its solution graph is isomorphic to G after certain adjacent vertices are merged. In addition to being an interesting structural result in its own right, the Faithful Expressibility Theorem implies that all non-tight relations have the same

[1] A Boolean relation can be thought of as a template for a clause, see Section2 for precise definitions

computational complexity for both the connectivity and the st-connectivity problems. It also shows that the diameter of the solution graph of formulas obtainable such relations are polynomially related.

As a consequence of the Faithful Expressibility Theorem we establish three dichotomy results. The first is a dichotomy theorem for the st-connectivity problem: Given a Boolean formula φ and two solutions $\mathbf{s}$ and $\mathbf{t}$ of φ, is there a path from $\mathbf{s}$ to $\mathbf{t}$ in $G(\varphi)$? We show that st-connectivity is solvable in linear time for formulas built from tight relations, and PSPACE-complete in all other cases. The second is a dichotomy theorem for the connectivity problem: Given a Boolean formula φ, is $G(\varphi)$ connected? We show that connectivity is in coNP for formulas built from tight relations, and PSPACE-complete in all other cases. Finally, we establish a structural dichotomy theorem for the diameter of the connected components of the solution space of Boolean formulas. This result asserts that, in the PSPACE-complete cases, the diameter of the connected components can be exponential, but in all other cases it is linear.

Technical Contributions. In Schaefer's Dichotomy Theorem, NP-hardness of satisfiability was a consequence of an expressibility theorem, which asserted that every Boolean relation can be obtained as a projection over a formula built from clauses from any "hard" set of relations (i.e. a set in which at least one relation is not bijunctive, at least one is not Horn, at least one is not dual Horn, and at least one is not affine). Schaefer's notion of expressibility is inadequate for our problem, so we introduce and work with a delicate and more strict notion of expressibility, which we call *faithful expressibility*. Intuitively, faithful expressibility means that, in addition to definability via a projection, the space of witnesses of the existential quantifiers in the projection has certain strong connectivity properties that allow us to capture the graph structure of the relation that is being defined. It should be noted that Schaefer's Dichotomy Theorem can also be proved using a Galois connection and Post's celebrated classification of the lattice of Boolean clones (see [4]). This method, however, does not appear to apply to connectivity, as the boundaries discovered here cut across Boolean clones. Thus, the use of faithful expressibility or some other refined definability technique seems unavoidable.

The main technical challenge in this work is the proof of the Faithful Expressibility Theorem, which is proved via a series of reductions. To prove it, we identify the simplest non-tight relations: these are ternary relations whose graph is a path of length 4 between assignments at Hamming distance 2. We show that one can faithfully express such a path from any non-tight set of relations. Next, we show that these paths can faithfully express all 3-CNF clauses, which are then easily shown to faithfully express any relation.

The Faithful Expressibility Theorem allows us to focus on a specific non-tight set of relations in order to establish the hard part of our dichotomies, We show that both connectivity and st-connectivity are hard for 3-CNF formulas; this is proved by a reduction from a generic PSPACE computation. Similarly, we show that formulas built from non-tight relations can have large diameter by explicitly constructing a 3-CNF formula on n variables whose diameter is exponential in n.

Our upper bounds for tight sets of relations are proved using structural properties that characterize the solution graphs. For tight sets of relation, we show that every component has a unique minimum element, or every component has a unique maximum element, or the Hamming distance coincides with the shortest-path distance in the relation. These properties are inherited by every formula built from a tight set of relations, and yield both small diameter and linear algorithms for st-connectivity.

An intriguing byproduct of our work is that we have identified a broad class of NP-complete satisfiability problems - those built from tight relations - that have simple structural properties, such as linear diameter. It would be interesting to investigate if these properties make random instances built from tight relations easier for WalkSAT and similar heuristics, and if so, whether such heuristics are amenable to rigorous analysis.

Organization of this Paper. In Section 2 we introduce the main concepts precisely, and state our results. We prove the two sides of the dichotomy in Sections 3 and 4 respectively. Finally, we will discuss a few open questions and conjectures in Section 5. An extended abstract of this paper appears in ICALP'06 [10].

2 Basic Concepts and Statements of Results

A *logical relation* R is a non-empty subset of $\{0,1\}^k$, for some $k \geq 1$; k is the *arity* of R. Let $\mathcal{S}$ be a finite set of logical relations. A CNF$(\mathcal{S})$*-formula* over a set of variables $V = \{x_1, \ldots, x_n\}$ is a finite conjunction $C_1 \wedge \cdots \wedge C_n$ of clauses built using relations from $\mathcal{S}$, variables from V, and the constants 0 and 1; this means that each C_i is an expression of the form $R(\xi_1, \ldots, \xi_k)$, where $R \in \mathcal{S}$ is a relation of arity k, and each ξ_j is a variable in V or one of the constants 0, 1.

The *satisfiability problem* SAT$(\mathcal{S})$ associated with a finite set $\mathcal{S}$ of logical relations asks: given a CNF$(\mathcal{S})$-formula φ, is it satisfiable? All well known restrictions of Boolean satisfiability, such as 3-SAT, NOT-ALL-EQUAL 3-SAT, and POSITIVE 1-IN-3 SAT, can be cast as SAT$(\mathcal{S})$ problems, for a suitable choice of $\mathcal{S}$. For instance, POSITIVE 1-IN-3SAT is SAT$(\{R_{1/3}\})$, where $R_{1/3} = \{100, 010, 001\}$. Schaefer [20] identified the complexity of *every* satisfiability problem SAT$(\mathcal{S})$. To state Schaefer's main result, we need to define some basic concepts.

Definition 1 Let R be a logical relation.

1. R is *bijunctive* if it is the set of solutions of a 2-CNF formula.

2. R is *Horn* if it is the set of solutions of a Horn formula, where a Horn formula is a CNF formula such that each conjunct has at most one positive literal.

3. R is *dual Horn* if it is the set of solutions of a dual Horn formula, where a dual Horn formula is a CNF formula such that each conjunct has at most one negative literal.

4. R is *affine* if it is the set of solutions of a system of linear equations over $\mathbb{Z}_2$.

Each of these types of logical relations can be characterized in terms of *closure* properties [20]. A relation R is bijunctive if and only if it is closed under the *majority* operation (if $\mathbf{a}, \mathbf{b}, \mathbf{c} \in R$, then $\mathrm{maj}(\mathbf{a}, \mathbf{b}, \mathbf{c}) \in R$, where $\mathrm{maj}(\mathbf{a}, \mathbf{b}, \mathbf{c})$ is the vector whose i-th bit is the majority of a_i, b_i, c_i). A relation R is Horn if and only if it is closed under $\vee$ (if $\mathbf{a}, \mathbf{b} \in R$, then $\mathbf{a} \vee \mathbf{b} \in R$, where, $\mathbf{a} \vee \mathbf{b}$ is the vector whose i-th bit is $a_i \vee b_i$). Similarly, R is dual Horn if and only if it is closed under $\wedge$. Finally, R is affine if and only if it is closed under $\mathbf{a} \oplus \mathbf{b} \oplus \mathbf{c}$.

Definition 2 A set $\mathcal{S}$ of logical relations is *Schaefer* if at least one of the following 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.

Theorem 1 (Schaefer's Dichotomy Theorem [20]) *If $\mathcal{S}$ is Schaefer, then* SAT$(\mathcal{S})$ *is in* P*; otherwise,* SAT$(\mathcal{S})$ *is* NP*-complete.*

Note that the closure properties of Schaefer sets yield a cubic algorithm for determining, given a finite set $\mathcal{S}$ of relations, whether SAT$(\mathcal{S})$ is in P or NP-complete (the input size is the sum of the sizes of relations in $\mathcal{S}$).

Here, we are interested in the connectivity properties of the space of solutions of CNF$(\mathcal{S})$-formulas. If φ is a CNF$(\mathcal{S})$-formula with n variables, then $G(\varphi)$ denotes the subgraph of the n-dimensional hypercube induced by the solutions of φ. Thus, the vertices of $G(\varphi)$ are the solutions of φ, and there is an edge between two solutions of $G(\varphi)$ precisely when they differ in a single variable.

We consider the following two algorithmic problems for CNF$(\mathcal{S})$-formulas.

1. The *connectivity* problem $\text{CONN}(\mathcal{S})$: given a $\text{CNF}(\mathcal{S})$-formula φ, is $G(\varphi)$ connected?

2. The *st-connectivity* problem $\text{ST-CONN}(\mathcal{S})$: given a $\text{CNF}(\mathcal{S})$-formula φ and two solutions $\mathbf{s}$ and $\mathbf{t}$ of φ, is there a path from $\mathbf{s}$ to $\mathbf{t}$ in $G(\varphi)$?

To pinpoint the computational complexity of $\text{ST-CONN}(\mathcal{S})$ and $\text{CONN}(\mathcal{S})$, we need to introduce certain new types of relations.

Definition 3 Let $R \subseteq \{0,1\}^k$ be a logical relation.

1. R is *componentwise bijunctive* if every connected component of $G(R)$ is bijunctive.

2. R is OR-*free* if the relation $\text{OR} = \{01, 10, 11\}$ cannot be obtained from R by setting $k-2$ of the coordinates of R to a constant $\mathbf{c} \in \{0,1\}^{k-2}$. In other words, R is OR-free if $(x_1 \vee x_2)$ is not definable from R by fixing $k-2$ variables.

3. R is NAND-*free* if $(\bar{x}_1 \vee \bar{x}_2)$ is not definable from R by fixing $k-2$ variables.

The next lemma follows from the closure properties of bijunctive, Horn, and dual Horn relations.

Lemma 1 *Let R be a logical relation.*

1. *If R is bijunctive, then R is componentwise bijunctive.*

2. *If R is Horn, then R is* OR-*free.*

3. *If R is dual Horn, then R is* NAND-*free.*

4. *If R is affine, then R is componentwise bijunctive,* OR-*free, and* NAND-*free.*

These containments are proper. For instance, $R_{1/3} = \{100, 010, 001\}$ is componentwise bijunctive, but not bijunctive as $\text{maj}(100, 010, 001) = 000 \notin R_{1/3}$.

We are now ready to introduce the key concept of a *tight* set of relations.

Definition 4 A set $\mathcal{S}$ of logical relations is *tight* if at least one of the following three conditions holds:

1. Every relation in $\mathcal{S}$ is componentwise bijunctive;

2. Every relation in $\mathcal{S}$ is OR-free;

3. Every relation in $\mathcal{S}$ is NAND-free.

In view of Lemma 1, if $\mathcal{S}$ is Schaefer, then it is tight. The converse, however, does not hold. It is also easy to see that there is a polynomial-time algorithm for testing whether a given finite set $\mathcal{S}$ of logical relations is tight.

The main step in the proof of Schaefer's dichotomy theorem is a result known as Schaefer's expressibility theorem. Similarly, the crux of our results is the following theorem which we will call the Faithful Expressibility Theorem. At a high level, this theorem asserts that for any Boolean relation with a solution graph G, we can construct a formula using any non-tight set of relations, such that its solution graph is isomorphic to G after certain adjacent vertices are merged. See section 4 for a precise definition of faithful expressibility.

Theorem 2 (Faithful Expressibility Theorem) *Let $\mathcal{S}$ be a set of relations that is not tight. Every relation is faithfully expressible from $\mathcal{S}$.*

Using the Faithful Expressibility Theorem, we obtain dichotomy theorems for the computational complexity of $\text{CONN}(\mathcal{S})$ and $\text{ST-CONN}(\mathcal{S})$.

Theorem 3 *Let $\mathcal{S}$ be a finite set of logical relations. If $\mathcal{S}$ is tight, then* $\text{CONN}(\mathcal{S})$ *is in* coNP*; otherwise, it is* PSPACE-*complete.*

Theorem 4 *Let $\mathcal{S}$ be a finite set of logical relations. If $\mathcal{S}$ is tight, then* $\text{ST-CONN}(\mathcal{S})$ *is in* P*; otherwise,* $\text{ST-CONN}(\mathcal{S})$ *is* PSPACE-*complete.*

We also show that if $\mathcal{S}$ is tight, but not Schaefer, then $\text{CONN}(\mathcal{S})$ is coNP-complete.

The dichotomy in the computational complexity of $\text{CONN}(\mathcal{S})$ and $\text{ST-CONN}(\mathcal{S})$ is accompanied by a parallel structural dichotomy in the size of the diameter of $G(\varphi)$ (where, for a $\text{CNF}(\mathcal{S})$-formula φ, the *diameter of* $G(\varphi)$ is the maximum of the diameters of the components of $G(\varphi)$).

Theorem 5 *Let $\mathcal{S}$ be a finite set of logical relations. If $\mathcal{S}$ is tight, then for every* $\text{CNF}(\mathcal{S})$*-formula φ, the diameter of $G(\varphi)$ is linear in the number of variables of φ; otherwise, there are* $\text{CNF}(\mathcal{S})$*-formulas φ such that the diameter of $G(\varphi)$ is exponential in the number of variables of φ.*

Our results and their comparison to Schaefer's Dichotomy Theorem are summarized in the table below.

$\mathcal{S}$	$\text{SAT}(\mathcal{S})$	$\text{ST-CONN}(\mathcal{S})$	$\text{CONN}(\mathcal{S})$	Diameter
Schaefer	P	P	coNP	$O(n)$
Tight, non-Schaefer	NP-complete	P	coNP-complete	$O(n)$
Non-tight	NP-complete	PSPACE-complete	PSPACE-complete	$2^{\Omega(\sqrt{n})}$

As an example, the set $\mathcal{S} = \{R_{1/3}\}$, where $R_{1/3} = \{100, 010, 001\}$, is tight, but not Schaefer. It follows that $\text{SAT}(\mathcal{S})$ is NP-complete (recall that this problem is POSITIVE 1-IN-3 SAT), $\text{ST-CONN}(\mathcal{S})$ is in P, and $\text{CONN}(\mathcal{S})$ is coNP-complete. Consider also the set $\mathcal{S} = \{R_{\text{NAE}}\}$, where $R_{\text{NAE}} = \{0,1\}^3 \setminus \{000, 111\}$. This set is not tight, hence $\text{SAT}(\mathcal{S})$ is NP-complete (this problem is POSITIVE NOT-ALL-EQUAL 3-SAT), while both $\text{ST-CONN}(\mathcal{S})$ and $\text{CONN}(\mathcal{S})$ are PSPACE-complete.

We conjecture that if $\mathcal{S}$ is Schaefer, then $\text{CONN}(\mathcal{S})$ is in P. If this conjecture is true, it will follow that the complexity of $\text{CONN}(\mathcal{S})$ exhibits a *trichotomy*: if $\mathcal{S}$ is Schaefer, then $\text{CONN}(\mathcal{S})$ is in P; if $\mathcal{S}$ is tight, but not Schaefer, then $\text{CONN}(\mathcal{S})$ is coNP-complete; if $\mathcal{S}$ is not tight, then $\text{CONN}(\mathcal{S})$ is PSPACE-complete.

3 The Easy Case of the Dichotomy: Tight Sets of Relations

In this section, we explore some structural properties for the solution graphs of tight sets of relations. These properties provide simple algorithms for $\text{CONN}(\mathcal{S})$ and $\text{ST-CONN}(\mathcal{S})$ for tight sets $\mathcal{S}$, and also guarantee that for such sets, the diameter of $G(\varphi)$ of $\text{CNF}(\mathcal{S})$-formula φ is linear.

We will use $\mathbf{a}, \mathbf{b}, \ldots$ to denote Boolean vectors, and $\mathbf{x}$ and $\mathbf{y}$ to denote vectors of variables. We write $|\mathbf{a}|$ to denote the Hamming weight (number of 1's) of a Boolean vector $\mathbf{a}$. Given two Boolean vectors $\mathbf{a}$ and $\mathbf{b}$, we write $|\mathbf{a} - \mathbf{b}|$ to denote the Hamming distance between $\mathbf{a}$ and $\mathbf{b}$. Finally, if $\mathbf{a}$ and $\mathbf{b}$ are solutions of a Boolean formula φ and lie in the same component of $G(\varphi)$, then we write $d_\varphi(\mathbf{a}, \mathbf{b})$ to denote the shortest-path distance between $\mathbf{a}$ and $\mathbf{b}$ in $G(\varphi)$.

3.1 Componentwise Bijunctive Sets of Relations

Lemma 2 *Let $\mathcal{S}$ be a set of componentwise bijunctive relations and φ a* CNF($\mathcal{S}$)*-formula. If $\mathbf{a}$ and $\mathbf{b}$ are two solutions of φ that lie in the same component of $G(\varphi)$, then $d_\varphi(\mathbf{a}, \mathbf{b}) = |\mathbf{a} - \mathbf{b}|$.*

PROOF: Consider first the special case in which every relation in $\mathcal{S}$ is bijunctive. In this case, φ is equivalent to a 2-CNF formula and so the space of solutions of φ is closed under majority. We show that there is a path in $G(\varphi)$ from $\mathbf{a}$ to $\mathbf{b}$, such that along the path only the assignments on variables with indices from the set $D = \{i : a_i \neq b_i\}$ change. This implies that the shortest path is of length $|D|$ by induction on $|D|$. Consider any path $\mathbf{a} \to \mathbf{u^1} \to \cdots \to \mathbf{u^r} \to \mathbf{b}$ in $G(\varphi)$. We construct another path by replacing $\mathbf{u^i}$ by $\mathbf{v^i} = \mathrm{maj}\ (\mathbf{a}, \mathbf{u^i}, \mathbf{b})$ for $i = 1, \ldots, r$, and removing repetitions. This is a path because for any i $\mathbf{v^i}$ and $\mathbf{v^{i+1}}$ differ in at most one variable. Furthermore, $\mathbf{v^i}$ agrees with $\mathbf{a}$ and $\mathbf{b}$ for every i for which $a_i = b_i$. Therefore, along this path only variables in D are flipped.

For the general case, we show that every component F of $G(\varphi)$ is the solution space of a 2-CNF formula φ'. Let F be the component of $G(\varphi)$ which contains $\mathbf{a}$ and $\mathbf{b}$. Let $R \in \mathcal{S}$ be a relation with two components, R_1, R_2 each of which are bijunctive. Consider a clause in φ of the form $R(x_1, \ldots, x_k)$. The projection of F onto $x_1, \ldots, x_k$ is itself connected and must satisfy R. Hence it lies within one of the two components R_1, R_2, assume it is R_1. We replace $R(x_1, \ldots, x_k)$ by $R_1(x_1, \ldots, x_k)$. Call this new formula φ_1. $G(\varphi_1)$ consists of all components of $G(\varphi)$ whose projection on $x_1, \ldots, x_k$ lies in R_1. We repeat this for every clause. Finally we are left with a formula φ' over a set of bijunctive relations. Hence φ' is bijunctive and $G(\varphi')$ is a component of $G(\varphi)$. So the claim follows from the bijunctive case. □

Corollary 1 *Let $\mathcal{S}$ be a set of componentwise bijunctive relations. Then*

1. *For every $\varphi \in$ CNF($\mathcal{S}$) with n variables, the diameter of each component of $G(\varphi)$ is bounded by n.*
2. ST-CONN($\mathcal{S}$) *is in* P.
3. CONN($\mathcal{S}$) *is in* coNP.

PROOF: The bound on diameter is an immediate consequence of Lemma 2.

The following algorithm solves ST-CONN($\mathcal{S}$) given vertices $\mathbf{s}, \mathbf{t} \in G(\varphi)$. Start with $\mathbf{u} = \mathbf{s}$. At each step, find a variable x_i so that $u_i \neq t_i$ and flip it, until we reach $\mathbf{t}$. If at any stage no such variable exists, then declare that $\mathbf{s}$ and $\mathbf{t}$ are not connected. If the $\mathbf{s}$ and $\mathbf{t}$ are disconnected, the algorithm is bound to fail. So assume that they are connected. Correctness is proved by induction on $d = |\mathbf{s} - \mathbf{t}|$. It is clear that the algorithm works when $d = 1$. Assume that the algorithm works for $d - 1$. If s and t are connected and are distance d apart, Lemma 2 implies there is a path of length d between them in $G(\varphi)$. In particular, the algorithm will find a variable x_i to flip. The resulting assignment is at distance $d - 1$ from $\mathbf{t}$, so now we proceed by induction.

Next we prove that CONN($\mathcal{S}$) $\in$ coNP. A short certificate that the graph is not connected is a pair of assignments $\mathbf{s}$ and $\mathbf{t}$ which are solutions from different components. To verify that they are disconnected it suffices to run the algorithm for ST-CONN. □

3.2 OR-free and NAND-free Sets of Relations

We consider sets of OR-free relations. Sets of NAND-free relations are handled dually. Define the *coordinate-wise partial order* $\leq$ on Boolean vectors as follows: $\mathbf{a} \leq \mathbf{b}$ if $a_i \leq b_i$, for each i.

Lemma 3 *Let $\mathcal{S}$ be a set of* OR*-free relations and φ a* CNF($\mathcal{S}$)*-formula. Every component of $G(\varphi)$ contains a minimum solution with respect to the coordinate-wise order; moreover, every solution is connected to the minimum solution in the same component via a monotone path.*

PROOF: We call a satisfying assignment locally minimal, if it has no neighboring satisfying assignments that are smaller than it. We will show that there is exactly one such assignment in each component of $G(\varphi)$.

Suppose there are two distinct locally minimal assignments $\mathbf{u}$ and $\mathbf{u'}$ in some component of $G(\varphi)$. Consider the path between them where the maximum Hamming weight of assignments on the path is minimized. If there are many such paths, pick one where the smallest number of assignments have the maximum Hamming weight. Denote this path by $\mathbf{u} = \mathbf{u^1} \to \mathbf{u^2} \to \cdots \to \mathbf{u^r} = \mathbf{u'}$. Let $\mathbf{u^i}$ be an assignment of largest Hamming weight in the path. Then $\mathbf{u^i} \neq \mathbf{u}$ and $\mathbf{u^i} \neq \mathbf{u'}$, since $\mathbf{u}$ and $\mathbf{u'}$ are locally minimal. The assignments $\mathbf{u^{i-1}}$ and $\mathbf{u^{i+1}}$ differ in exactly 2 variables, say, in x_1 and x_2. So $\{u_1^{i-1}u_2^{i-1},\ u_1^i u_2^i,\ u_1^{i+1}u_2^{i+1}\} = \{01, 11, 10\}$. Let $\hat{\mathbf{u}}$ be such that $\hat{u}_1 = \hat{u}_2 = 0$, and $\hat{u}_i = u_i$ for $i > 2$. If $\hat{\mathbf{u}}$ is a solution, then the path $\mathbf{u^1} \to \mathbf{u^2} \to \cdots \to \mathbf{u^i} \to \hat{\mathbf{u}} \to \mathbf{u^{i+1}} \to \cdots \to \mathbf{u^r}$ contradicts the way we chose the original path. Therefore, $\hat{\mathbf{u}}$ is not a solution. This means that there is a clause that is violated by it, but is satisfied by $\mathbf{u^{i-1}}$, $\mathbf{u^i}$, and $\mathbf{u^{i+1}}$. So the relation corresponding to that clause is not OR-free, which is a contradiction.

The unique locally minimal solution in a component is its minimum solution, because starting from any other assignment in the component, it is possible to keep moving to neighbors that are smaller, and the only time it becomes impossible to find such a neighbor is when the locally minimal solution is reached. Therefore, there is a monotone path from any satisfying assignment to the minimum in that component. □

Corollary 2 *Let $\mathcal{S}$ be a set of* OR*-free relations. Then*

1. *For every $\varphi \in$ CNF$(\mathcal{S})$ with n variables, the diameter of each component of $G(\varphi)$ is bounded by $2n$.*
2. ST-CONN$(\mathcal{S})$ *is in* P.
3. CONN$(\mathcal{S})$ *is in* coNP.

PROOF: Given solutions $\mathbf{s}$ and $\mathbf{t}$ in the same component of $G(\varphi)$, there is a monotone path from each to the minimal solution $\mathbf{u}$ in the component. This gives a path from $\mathbf{s}$ to $\mathbf{t}$ of length at most $2n$. To check if $\mathbf{s}$ and $\mathbf{t}$ are connected, we just check that the minimal assignments reached from $\mathbf{s}$ and $\mathbf{t}$ are the same. □

3.3 The Complexity of CONN$(\mathcal{S})$ for Tight Sets of Relations

We can further specify the complexity of CONN$(\mathcal{S})$ for the tight cases which are not Schaefer, using a result of Juban [12].

Lemma 4 *For $\mathcal{S}$ tight, but not Schaefer,* CONN$(\mathcal{S})$ *is* coNP*-complete.*

PROOF: The problem ANOTHER-SAT$(\mathcal{S})$ is: given a formula φ in CNF$(\mathcal{S})$ and a solution $\mathbf{s}$, does there exist a solution $\mathbf{t} \neq \mathbf{s}$? Juban ([12], Theorem 2) shows that if $\mathcal{S}$ is not Schaefer, then ANOTHER-SAT is NP-complete. He also shows ([12], Corollary 1) that if $\mathcal{S}$ is not Schaefer, then the relation $x \neq y$ is expressible from $\mathcal{S}$ through substitutions.

Since $\mathcal{S}$ is not Schaefer, ANOTHER-SAT$(\mathcal{S})$ is NP-complete. Let $\varphi, \mathbf{s}$ be an instance of ANOTHER-SAT on variables $x_1, \ldots, x_n$. We define a CNF$(\mathcal{S})$ formula ψ on $x_1, \ldots, x_n, y_1, \ldots, y_n$ as

$$\psi(x_1, \ldots, x_n, y_1, \ldots, y_n) = \varphi(x_1, \ldots, x_n) \wedge_i (x_i \neq y_i)$$

It is easy to see that $G(\psi)$ is connected if and only if $\mathbf{s}$ is the unique solution to φ. □

Further we can show that CONN$(\mathcal{S})$ is in P if $\mathcal{S}$ is affine or bijunctive. Thus the only tight cases for which CONN$(\mathcal{S})$ is not known to be coNP-complete or in P are Horn and dual-Horn. We conjecture that these problems are in P.

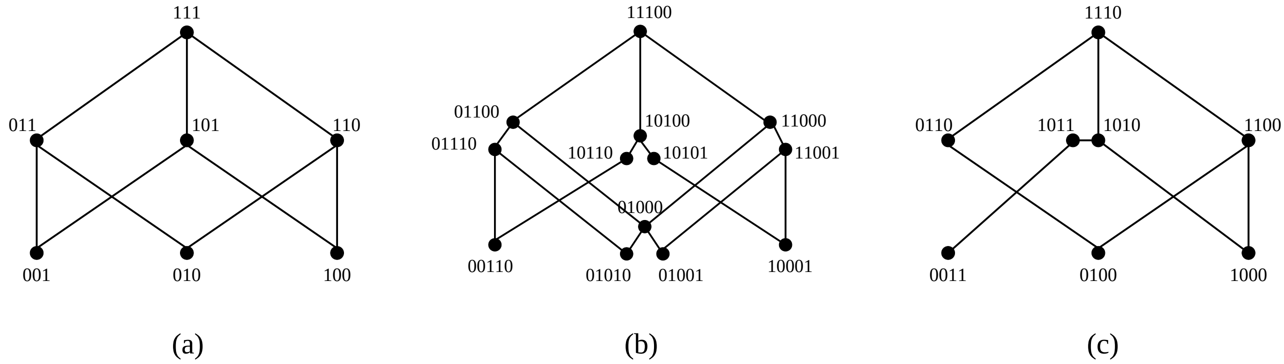

Figure 1: Expressing the relation $(x_1 \vee x_2 \vee x_3)$ using NOT-ALL-EQUAL relations.
(a) The graph of $(x_1 \vee x_2 \vee x_3)$;
(b) The graph of a faithful expression: $\varphi(\mathbf{x}, y_1, y_2) = R_{\text{NAE}}(x_1, x_2, y_1) \wedge R_{\text{NAE}}(x_2, x_3, y_2) \wedge R_{\text{NAE}}(y_1, y_2, 1)$.
(c) The graph of an unfaithful expression: $\varphi(\mathbf{x}, y_1) = R_{\text{NAE}}(x_1, x_2, y_1) \wedge R_{\text{NAE}}(\bar{y}_1, x_3, 0) \wedge R_{\text{NAE}}(y_1, x_2, 1)$.
In both cases $(x_1 \vee x_2 \vee x_3) = \exists \mathbf{y}\ \varphi(\mathbf{x}, \mathbf{y})$, but only in the first case the connectivity is preserved.

4 The Hard Case of the Dichotomy: Non-Tight Sets of Relations

We will show that all non-tight sets of relations lead to solution graphs that have identical properties in a natural sense that is captured in the notion of faithful expressibility. We define this notion in Section 4.1, and prove the Faithful Expressibility Theorem in Section 4.2. This theorem implies that the complexity of the connectivity questions for all such sets is the same, and the possible diameter of components of the solution graph is also related polynomially. In section 4.3 we will prove that for 3-CNF formulas the connectivity questions are PSPACE-complete, and the diameter can be exponential. This fact together with the Faithful Expressibility Theorem implies the hard side of all of our dichotomy results.

4.1 Faithful Expressibility

In his dichotomy theorem, Schaefer [20] used the following notion of expressibility: a relation R is *expressible from* a set $\mathcal{S}$ of relations if there is a CNF($\mathcal{S}$)-formula φ so that $R(\mathbf{x}) \equiv \exists \mathbf{y}\ \varphi(\mathbf{x}, \mathbf{y})$. This notion, is not sufficient for our purposes. Instead, we introduce a more delicate notion, which we call *faithful expressibility*. Intuitively, we view the relation R as a subgraph of the hypercube, rather than just a subset, and require that this graph structure be also captured by the formula φ.

Definition 5 *A relation R is* faithfully expressible *from a set of relations $\mathcal{S}$ if there is a* CNF($\mathcal{S}$)*-formula φ such that the following condition hold:*

1. *$R = \{\mathbf{a} :\ \exists \mathbf{y}\ \varphi(\mathbf{a}, \mathbf{y})\}$;*

2. *For every $\mathbf{a} \in R$, the graph $G(\varphi(\mathbf{a}, \mathbf{y}))$ is connected;*

3. *For $\mathbf{a}, \mathbf{b} \in R$ with $|\mathbf{a} - \mathbf{b}| = 1$, there exists $\mathbf{w}$ such that $(\mathbf{a}, \mathbf{w})$ and $(\mathbf{b}, \mathbf{w})$ are solutions of φ.*

For $\mathbf{a} \in R$, the *witnesses* of $\mathbf{a}$ are the $\mathbf{y}$'s such that $\varphi(\mathbf{a}, \mathbf{y})$ is true. The last two conditions say that the witnesses of $\mathbf{a} \in R$ are connected, and that neighboring $\mathbf{a}, \mathbf{b} \in R$ have a common witness. This allows us to simulate an edge $(\mathbf{a}, \mathbf{b})$ in $G(R)$ by a path in $G(\varphi)$, and thus relate the connectivity properties of the solution spaces. There is however, a price to pay: it is much harder to come up with formulas that faithfully express a relation R. An example is when $\mathcal{S}$ is the set of all paths of length 4 in $\{0,1\}^3$, a set that plays a crucial role in our proof. While 3-SAT relations are easily expressible from $\mathcal{S}$ in Schaefer's sense, the CNF($\mathcal{S}$)-formulas that faithfully express 3-SAT relations are fairly complicated and have a large witness space.

An example of the difference between a faithful and an unfaithful expression is shown in Figure 4.1.

Lemma 5 *Let $\mathcal{S}$ and $\mathcal{S}'$ be sets of relations such that every $R \in \mathcal{S}'$ is faithfully expressible from $\mathcal{S}$. Given a* CNF($\mathcal{S}'$)*-formula $\psi(\mathbf{x})$, one can efficiently construct a* CNF($\mathcal{S}$)*-formula $\varphi(\mathbf{x},\mathbf{y})$ such that:*

1. *$\psi(\mathbf{x}) \equiv \exists \mathbf{y}\ \varphi(\mathbf{x},\mathbf{y})$;*
2. *if $(\mathbf{s},\mathbf{w^s}),(\mathbf{t},\mathbf{w^t}) \in \varphi$ are connected in $G(\varphi)$ by a path of length d, then there is a path from $\mathbf{s}$ to $\mathbf{t}$ in $G(\psi)$ of length at most d;*
3. *If $\mathbf{s},\mathbf{t} \in \psi$ are connected in $G(\psi)$, then for every witness $\mathbf{w^s}$ of $\mathbf{s}$, and every witness $\mathbf{w^t}$ of $\mathbf{t}$, there is a path from $(\mathbf{s},\mathbf{w^s})$ to $(\mathbf{t},\mathbf{w^t})$ in $G(\varphi)$.*

PROOF: Suppose ψ is a formula on n variables that consists of m clauses $C_1,\ldots,C_m$. For clause C_j, assume that the set of variables is $V_j \subseteq [n]$, and that it involves relation $R_j \in \mathcal{S}$. Thus, $\psi(\mathbf{x})$ is $\wedge_{j=1}^m R_j(\mathbf{x}_{V_j})$. Let φ_j be the faithful expression for R_j from $\mathcal{S}'$, so that $R_j(\mathbf{x}_{V_j}) \equiv \exists \mathbf{y}_j\ \varphi_j(\mathbf{x}_{V_j},\mathbf{y}_j)$. Let $\mathbf{y}$ be the vector $(\mathbf{y}_1,\ldots,\mathbf{y}_m)$ and let $\varphi(\mathbf{x},\mathbf{y})$ be the formula $\wedge_{j=1}^m \varphi_j(\mathbf{x}_{V_j},\mathbf{y}_j)$. Then $\psi(\mathbf{x}) \equiv \exists \mathbf{y}\ \varphi(\mathbf{x},\mathbf{y})$.

Statement (2) follows from (1) by projection of the path on the coordinates of $\mathbf{x}$. For statement (3), consider $\mathbf{s},\mathbf{t} \in \psi$ that are connected in $G(\psi)$ via a path $\mathbf{s} = \mathbf{u^0} \to \mathbf{u^1} \to \cdots \to \mathbf{u^r} = \mathbf{t}$. For every $\mathbf{u^i},\mathbf{u^{i+1}}$, and clause C_j, there exists an assignment $\mathbf{w^i}_j$ to $\mathbf{y}_j$ such that both $(\mathbf{u^i}_{V_j},\mathbf{w^i}_j)$ and $(\mathbf{u^{i+1}}_{V_j},\mathbf{w^i}_j)$ are solutions of φ_j, by condition (2) of faithful expressibility. Thus $(\mathbf{u^i},\mathbf{w^i})$ and $(\mathbf{u^{i+1}},\mathbf{w^i})$ are both solutions of φ, where $\mathbf{w^i} = (\mathbf{w^i}_1,\ldots\mathbf{w^i}_m)$. Further, for every $\mathbf{u^i}$, the space of solutions of $\varphi(\mathbf{u^i},\mathbf{y})$ is the product space of the solutions of $\varphi_j(\mathbf{u^i}_{V_j},\mathbf{y}_j)$ over $j = 1,\ldots,m$. Since these are all connected by condition (3) of faithful expressibility, $G(\varphi(\mathbf{u^i},\mathbf{y}))$ is connected. The following describes a path from $(\mathbf{s},\mathbf{w^s})$ to $(\mathbf{t},\mathbf{w^t})$ in $G(\varphi)$: $(\mathbf{s},\mathbf{w^s}) \rightsquigarrow (\mathbf{s},\mathbf{w^0}) \to (\mathbf{u^1},\mathbf{w^0}) \rightsquigarrow (\mathbf{u^1},\mathbf{w^1}) \to \cdots \rightsquigarrow (\mathbf{u^{r-1}},\mathbf{w^{r-1}}) \to (\mathbf{t},\mathbf{w^{r-1}}) \rightsquigarrow (\mathbf{t},\mathbf{w^t})$. Here $\rightsquigarrow$ indicates a path in $G(\varphi(\mathbf{u^i},\mathbf{y}))$. ☐

Corollary 3 *Suppose $\mathcal{S}$ and $\mathcal{S}'$ are sets of relations such that every $R \in \mathcal{S}'$ is faithfully expressible from $\mathcal{S}$.*

1. *There are polynomial time reductions from* CONN($\mathcal{S}'$) *to* CONN($\mathcal{S}$)*, and from* ST-CONN($\mathcal{S}'$) *to* ST-CONN($\mathcal{S}$).
2. *Given a* CNF($\mathcal{S}'$)*-formula $\psi(\mathbf{x})$ with m clauses, one can efficiently construct a* CNF($\mathcal{S}$)*-formula $\varphi(\mathbf{x},\mathbf{y})$ such that the length of $\mathbf{y}$ is $O(m)$ and the diameter of the solution space does not decrease.*

4.2 The Faithful Expressibility Theorem

In this subsection, we prove the Faithful Expressibility Theorem. The main step in the proof is Lemma 6 which shows that if $\mathcal{S}$ is not tight, then we can faithfully express the 3-clause relations from the relations in $\mathcal{S}$. If $k \geq 2$, then a *k-clause* is a disjunction of k variables or negated variables. For $0 \leq i \leq k$, let D_i be the set of all satisfying truth assignments of the k-clause whose first i literals are negated, and let $\mathcal{S}_k = \{D_0, D_1, \ldots, D_k\}$. Thus, CNF($\mathcal{S}_k$) is the collection of k-CNF formulas.

Lemma 6 *If set $\mathcal{S}$ of relations is not tight, $\mathcal{S}_3$ is faithfully expressible from $\mathcal{S}$.*

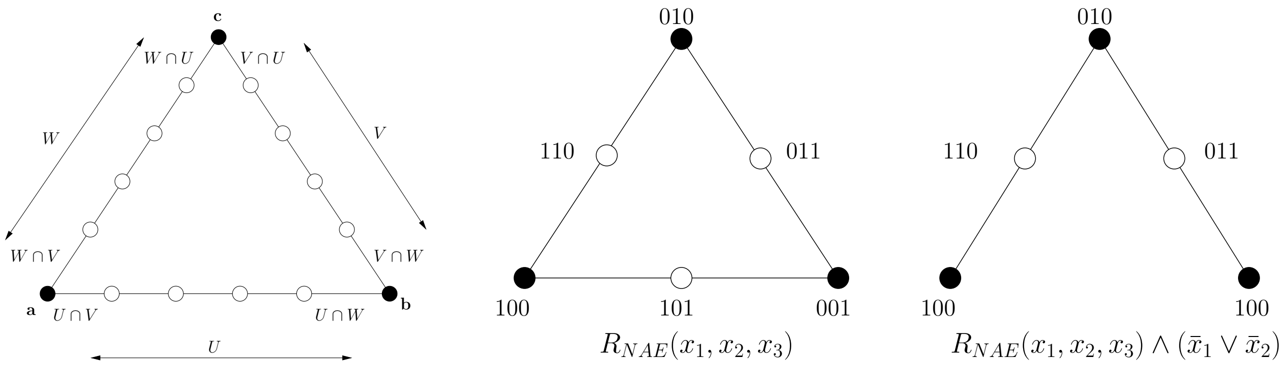

Figure 2: Proof of Step 1 of Lemma 6, and an example.

PROOF: First, observe that all 2-clauses are faithfully expressible from $\mathcal{S}$. There exists $R \in \mathcal{S}$ which is not OR-free, so we can express $(x_1 \vee x_2)$ by substituting constants in R. Similarly, we can express $(\bar{x}_1 \vee \bar{x}_2)$ using a relation that is not NAND-free. The last 2-clause $(x_1 \vee \bar{x}_2)$ can be obtained from OR and NAND by a technique that corresponds to reverse resolution. $(x_1 \vee \bar{x}_2) = \exists y\ (x_1 \vee y) \wedge (\bar{y} \vee \bar{x}_2)$. It is easy to see that this gives a faithful expression. From here onwards we assume that $\mathcal{S}$ contains all 2-clauses. The proof now proceeds in four steps. First, we will express a relation in which there exist two elements that are at graph distance larger than their Hamming distance. Second, we will express a relation that is just a single path between such elements. Third, we will express a relation which is a path of length 4 between elements at Hamming distance 2. Finally, we will express the 3-clauses.

Step 1 *Faithfully expressing a relation in which some distance expands.*

For a relation R, we say that the distance between $\mathbf{a}$ and $\mathbf{b}$ *expands* if $\mathbf{a}$ and $\mathbf{b}$ are connected in $G(R)$, but $d_R(\mathbf{a}, \mathbf{b}) > |\mathbf{a} - \mathbf{b}|$. By Lemma 2 no distance expands in componentwise bijunctive relations. This property also holds for the relation $R_{\mathrm{NAE}} = \{0, 1\}^3 \setminus \{000, 111\}$, which is not componentwise bijunctive. However, we show that if R is not componentwise bijunctive, then, by adding 2-clauses, we can faithfully express a relation Q in which some distance expands. For instance, when $R = R_{\mathrm{NAE}}$, then we can take $Q(x_1, x_2, x_3) = R_{\mathrm{NAE}}(x_1, x_2, x_3) \wedge (\bar{x_1} \vee \bar{x}_3)$. The distance between $\mathbf{a} = 100$ and $\mathbf{b} = 001$ in Q expands. Similarly, in the general construction, we identify $\mathbf{a}$ and $\mathbf{b}$ on a cycle, and add 2-clauses that eliminate all the vertices along the shorter arc between $\mathbf{a}$ and $\mathbf{b}$.

Since $\mathcal{S}$ is not tight, it contains a relation R which is not componentwise bijunctive. If R contains $\mathbf{a}, \mathbf{b}$ where the distance between them expands, we are done. So assume that for all $\mathbf{a}, \mathbf{b} \in G(R)$, $d_R(\mathbf{a}, \mathbf{b}) = |\mathbf{a} - \mathbf{b}|$. Since R is not componentwise bijunctive, there exists a triple of assignments $\mathbf{a}, \mathbf{b}, \mathbf{c}$ lying in the same component such that $\mathrm{maj}(\mathbf{a}, \mathbf{b}, \mathbf{c})$ is not in that component (which also easily implies it is not in R). Choose the triple such that the sum of pairwise distances $d_R(\mathbf{a}, \mathbf{b}) + d_R(\mathbf{b}, \mathbf{c}) + d_R(\mathbf{c}, \mathbf{a})$ is minimized. Let $U = \{i | a_i \neq b_i\}$, $V = \{i | b_i \neq c_i\}$, and $W = \{i | c_i \neq a_i\}$. Since $d_R(\mathbf{a}, \mathbf{b}) = |\mathbf{a} - \mathbf{b}|$, a shortest path does not flip variables outside of U, and each variable in U is flipped exactly once. The same holds for V and W. We note some useful properties of the sets U, V, W.

1. *Every index $i \in U \cup V \cup W$ occurs in exactly two of U, V, W.*
   Consider going by a shortest path from $\mathbf{a}$ to $\mathbf{b}$ to $\mathbf{c}$ and back to $\mathbf{a}$. Every $i \in U \cup V \cup W$ is seen an even number of times along this path since we return to $\mathbf{a}$. It is seen at least once, and at most thrice, so in fact it occurs twice.

2. *Every pairwise intersection $U \cap V, V \cap W$ and $W \cap U$ is non-empty.*
Suppose the sets U and V are disjoint. From Property 1, we must have $W = U \cup V$. But then it is easy to see that $\mathrm{maj}(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \mathbf{b}$ which is in R. This contradicts the choice of $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

3. *The sets $U \cap V$ and $U \cap W$ partition the set U.*
By Property 1, each index of U occurs in one of V and W as well. Also since no index occurs in all three sets U, V, W this is in fact a disjoint partition.

4. *For each index $i \in U \cap W$, it holds that $\mathbf{a} \oplus \mathbf{e}_i \notin R$.*
Assume for the sake of contradiction that $\mathbf{a}' = \mathbf{a} \oplus \mathbf{e}_i \in R$. Since $i \in U \cap W$ we have simultaneously moved closer to both $\mathbf{b}$ and $\mathbf{c}$. Hence we have $d_R(\mathbf{a}', \mathbf{b}) + d_R(\mathbf{b}, \mathbf{c}) + d_R(\mathbf{c}, \mathbf{a}') < d_R(\mathbf{a}, \mathbf{b}) + d_R(\mathbf{b}, \mathbf{c}) + d_R(\mathbf{c}, \mathbf{a})$. Also $\mathrm{maj}(\mathbf{a}', \mathbf{b}, \mathbf{c}) = \mathrm{maj}(\mathbf{a}, \mathbf{b}, \mathbf{c}) \notin R$. But this contradicts our choice of $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

Property 4 implies that the shortest paths to $\mathbf{b}$ and $\mathbf{c}$ diverge at $\mathbf{a}$, since for any shortest path to $\mathbf{b}$ the first variable flipped is from $U \cap V$ whereas for a shortest path to $\mathbf{c}$ it is from $W \cap V$. Similar statements hold for the vertices $\mathbf{b}$ and $\mathbf{c}$. Thus along the shortest path from $\mathbf{a}$ to $\mathbf{b}$ the first bit flipped is from $U \cap V$ and the last bit flipped is from $U \cap W$. On the other hand, if we go from $\mathbf{a}$ to $\mathbf{c}$ and then to $\mathbf{b}$, all the bits from $U \cap W$ are flipped before the bits from $U \cap V$. We use this crucially to define Q. We will add a set of 2-clauses that enforce the following rule on paths starting at $\mathbf{a}$: *Flip variables from $U \cap W$ before variables from $U \cap V$.* This will eliminate all shortest paths from $\mathbf{a}$ to $\mathbf{b}$ since they begin by flipping a variable in $U \cap V$ and end with $U \cap W$. The paths from $\mathbf{a}$ to $\mathbf{b}$ via $\mathbf{c}$ survive since they flip $U \cap W$ while going from $\mathbf{a}$ to $\mathbf{c}$ and $U \cap V$ while going from $\mathbf{c}$ to $\mathbf{b}$. However all remaining paths have length at least $|\mathbf{a} - \mathbf{b}| + 2$ since they flip twice some variables not in U.

Take all pairs of indices $\{(i, j) | i \in U \cap W, j \in U \cap V\}$. The following conditions hold from the definition of U, V, W: $a_i = \bar{c}_i = \bar{b}_i$ and $a_j = c_j = \bar{b}_j$. Add the 2-clause C_{ij} asserting that the pair of variables $x_i x_j$ must take values in $\{a_i a_j, c_i c_j, b_i b_j\} = \{a_i a_j, \bar{a}_i a_j, \bar{a}_i \bar{a}_j\}$. The new relation is $Q = R \wedge_{i,j} C_{ij}$. Note that $Q \subset R$. We verify that the distance between $\mathbf{a}$ and $\mathbf{b}$ in Q expands. It is easy to see that for any $j \in U$, the assignment $\mathbf{a} \oplus \mathbf{e}_j \notin Q$. Hence there are no shortest paths left from $\mathbf{a}$ to $\mathbf{b}$. On the other hand, it is easy to see that $\mathbf{a}$ and $\mathbf{b}$ are still connected, since the vertex $\mathbf{c}$ is still reachable from both.

Step 2 *Isolating a pair of assignments whose distance expands.*

The relation Q obtained in Step 1 may have several disconnected components. This *cleanup* step isolates a single pair of assignments whose distance expands. By adding 2-clauses, we show that one can express a path of length $r + 2$ between assignments at distance r.

Take $\mathbf{a}, \mathbf{b} \in Q$ whose distance expands in Q and $d_Q(\mathbf{a}, \mathbf{b})$ is minimized. Let $U = \{i : a_i \neq b_i\}$, and $|U| = r$. Shortest paths between $\mathbf{a}$ and $\mathbf{b}$ have certain useful properties:

1. *Each shortest path flips every variable from U exactly once.*
Observe that each index $j \in U$ is flipped an odd number of times along any path from $\mathbf{a}$ to $\mathbf{b}$. Suppose it is flipped thrice along a shortest path. Starting at $\mathbf{a}$ and going along this path, let $\mathbf{b}'$ be the assignment reached after flipping j twice. Then the distance between $\mathbf{a}$ and $\mathbf{b}'$ expands, since j is flipped twice along a shortest path between them in Q. Also $d_Q(\mathbf{a}, \mathbf{b}') < d_Q(\mathbf{a}, \mathbf{b})$, contradicting the choice of $\mathbf{a}$ and $\mathbf{b}$.

2. *Every shortest path flips exactly one variable $i \notin U$.*
Since the distance between $\mathbf{a}$ and $\mathbf{b}$ expands, every shortest path must flip some variable $i \notin U$. Suppose it flips more than one such variable. Since $\mathbf{a}$ and $\mathbf{b}$ agree on these variables, each of them is flipped an even number of times. Let i be the first variable to be flipped twice. Let $\mathbf{b}'$ be the assignment reached after flipping i the second time. It is easy to verify that the distance between $\mathbf{a}$ and $\mathbf{b}'$ also expands, but $d_Q(\mathbf{a}, \mathbf{b}') < d_Q(\mathbf{a}, \mathbf{b})$.

3. *The variable $i \notin U$ is the first and last variable to be flipped along the path.* Assume the first variable flipped is not i. Let $\mathbf{a}'$ be the assignment reached along the path before we flip i the first time. Then $d_Q(\mathbf{a}', \mathbf{b}) < d_Q(\mathbf{a}, \mathbf{b})$. The distance between $\mathbf{a}'$ and $\mathbf{b}$ expands since the shortest path between them flips the variables i twice. This contradicts the choice of $\mathbf{a}$ and $\mathbf{b}$. Assume $j \in U$ is flipped twice. Then as before we get a pair $\mathbf{a}', \mathbf{b}'$ that contradict the choice of $\mathbf{a}, \mathbf{b}$.

Every shortest path between $\mathbf{a}$ and $\mathbf{b}$ has the following structure: first a variable $i \notin U$ is flipped to $\bar{a}_i$, then the variables from U are flipped in some order,finally the variable i is flipped back to a_i.

Different shortest paths may vary in the choice of $i \notin U$ in the first step and in the order in which the variables from U are flipped. Fix one such path $T \subseteq Q$. Assume that $U = \{1, \ldots, r\}$ and the variables are flipped in this order, and the additional variable flipped twice is $r + 1$. Denote the path by $\mathbf{a} \to \mathbf{u^0} \to \mathbf{u^1} \to \cdots \to \mathbf{u^r} \to \mathbf{b}$. Next we prove that we cannot flip the $r + 1^{th}$ variable at an intermediate vertex along the path.

4 *For $1 \leq j \leq r - 1$ the assignment $\mathbf{u^j} \oplus \mathbf{e_{r+1}} \notin Q$.*
Suppose that for some j, we have $\mathbf{c} = \mathbf{u^j} \oplus \mathbf{e_{r+1}} \in Q$. Then $\mathbf{c}$ differs from $\mathbf{a}$ on $\{1, \ldots, i\}$ and from $\mathbf{b}$ on $\{i + 1, \ldots, r\}$. The distance from $\mathbf{c}$ to at least one of $\mathbf{a}$ or $\mathbf{b}$ must expand, else we get a path from $\mathbf{a}$ to $\mathbf{b}$ through $\mathbf{c}$ of length $|\mathbf{a} - \mathbf{b}|$ which contradicts the fact that this distance expands. However $d_Q(\mathbf{a}, \mathbf{c})$ and $d_Q(\mathbf{b}, \mathbf{c})$ are strictly less than $d_Q(\mathbf{a}, \mathbf{b})$ so we get a contradiction to the choice of $\mathbf{a}, \mathbf{b}$.

We now construct the path of length $r + 2$. For all $i \geq r + 2$ we set $x_i = a_i$ to get a relation on $r + 1$ variables. Note that $\mathbf{b} = \bar{a}_1 \ldots \bar{a}_r a_{r+1}$. Take $i < j \in U$. Along the path T the variable i is flipped before j so the variables $x_i x_j$ take one of three values $\{a_i a_j, \bar{a}_i a_j, \bar{a}_i \bar{a}_j\}$. So we add a 2-clause C_{ij} that requires $x_i x_j$ to take one of these values and take $T = Q \wedge_{i,j} C_{ij}$. Clearly, every assignment along the path lies in T. We claim that these are the only solutions. To show this, take an arbitrary assignment $\mathbf{c}$ satisfying the added constraints. If for some $i < j \leq r$ we have $c_i = a_i$ but $c_j = \bar{a}_j$, this would violate C_{ij}. Hence the first r variables of $\mathbf{c}$ are of the form $\bar{a}_1 \ldots \bar{a}_i a_{i+1} \ldots a_r$ for $0 \leq i \leq r$. If $c_{r+1} = \bar{a}_{r+1}$ then $\mathbf{c} = \mathbf{u}^i$. If $c_{r+1} = a_{r+1}$ then $\mathbf{c} = \mathbf{u}^i \oplus \mathbf{e}_{r+1}$. By property 4 above, such a vector satisfies Q if and only if $i = 0$ or $i = r$, which correspond to $\mathbf{c} = \mathbf{a}$ and $\mathbf{c} = \mathbf{b}$ respectively.

Step 3 *Faithfully expressing paths of length* 4.

Let $\mathcal{P}$ denote the set of all ternary relations whose graph is a path of length 4 between two assignments at Hamming distance 2. Up to permutations of coordinates, there are 6 such relations. Each of them is the conjunction of a 3-clause and a 2-clause. For instance, the relation $M = \{100, 110, 010, 011, 001\}$ can be written as of $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3)$. (It is named so, because its graph looks like the letter 'M' on the cube.) These relations are "minimal" examples of relations that are not componentwise bijunctive. By projecting out intermediate variables from the path T obtained in Step 2, we faithfully express one of the relations in $\mathcal{P}$. We faithfully express other relations in $\mathcal{P}$ using this relation.

We will write all relations in $\mathcal{P}$ in terms of $M(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3)$, by negating variables. For example $M(\bar{x}_1, x_2, x_3) = (\bar{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_3) = \{000, 010, 110, 111, 101\}$.

Define the relation $P(x_1, x_{r+1}, x_2) = \exists x_3 \ldots x_r \; T(x_1, \ldots, x_{r+1})$. The table below listing all tuples in P and their witnesses, shows that the conditions for faithful expressibility are satisfied, and $P \in \mathcal{P}$.

x_1, x_2, x_{r+1}	$x_3, \ldots, x_r$
$a_1 a_2 a_{r+1}$	$a_3 \ldots a_r$
$a_1 a_2 \bar{a}_{r+1}$	$a_3 \ldots a_r$
$\bar{a}_1 a_2 \bar{a}_{r+1}$	$a_3 \ldots a_r$
$\bar{a}_1 \bar{a}_2 \bar{a}_{r+1}$	$a_3 \ldots a_k,\ \bar{a}_3 a_4 \ldots a_r,\ \bar{a}_3 \bar{a}_4 a_5 \ldots a_r\ \ldots \bar{a}_3 \bar{a}_4 \ldots \bar{a}_r$
$\bar{a}_1 \bar{a}_2 a_{r+1}$	$\bar{a}_3 \bar{a}_4 \ldots \bar{a}_r$

Let $P(x_1, x_2, x_3) = M(l_1, l_2, l_3)$, where l_i is one of $\{x_i, \bar{x}_i\}$. We can now use P and 2-clauses to express every other relation in $\mathcal{P}$. Given $M(l_1, l_2, l_3)$ every relation in $\mathcal{P}$ can be obtained by negating some subset of the variables. Hence it suffices to show that we can express faithfully $M(\bar{l}_1, l_2, l_3)$ and $M(l_1, \bar{l}_2, l_3)$ (M is symmetric in x_1 and x_3). In the following let λ denote one of the literals $\{y, \bar{y}\}$, such that it is $\bar{y}$ if and only if l_1 is $\bar{x}_1$.

$$\begin{aligned}
M(\bar{l}_1, l_2, l_3) &= (\bar{l}_1 \vee l_2 \vee l_3) \wedge (l_1 \vee \bar{l}_3) \\
&= \exists y\ (\bar{l}_1 \vee \bar{\lambda}) \wedge (\lambda \vee l_2 \vee l_3) \wedge (l_1 \vee \bar{l}_3) \\
&= \exists y\ (\bar{l}_1 \vee \bar{\lambda}) \wedge (\lambda \vee l_2 \vee l_3) \wedge (l_1 \vee \bar{l}_3) \wedge (\bar{\lambda} \vee \bar{l}_3) \\
&= \exists y\ (\bar{l}_1 \vee \bar{\lambda}) \wedge (l_1 \vee \bar{l}_3) \wedge M(\lambda, l_2, l_3) \\
&= \exists y\ (\bar{l}_1 \vee \bar{\lambda}) \wedge (l_1 \vee \bar{l}_3) \wedge P(y, x_2, x_3)
\end{aligned}$$

For the next expression let λ denote one of the literals $\{y, \bar{y}\}$, such that it is negated if and only if l_2 is $\bar{x}_2$.

$$\begin{aligned}
M(l_1, \bar{l}_2, l_3) &= (l_1 \vee \bar{l}_2 \vee l_3) \wedge (\bar{l}_1 \vee \bar{l}_3) \\
&= \exists y\ (l_1 \vee l_3 \vee \lambda) \wedge (\bar{\lambda} \vee \bar{l}_2) \wedge (\bar{l}_1 \vee \bar{l}_3) \\
&= \exists y\ (\bar{\lambda} \vee \bar{l}_2) \wedge M(l_1, \lambda, l_3) \\
&= \exists y\ (\bar{\lambda} \vee \bar{l}_2) \wedge P(x_1, y, x_3)
\end{aligned}$$

The above expressions are both based on resolution and it is easy to check that they satisfy the properties of faithful expressibility.

Step 4 *Faithfully expressing $\mathcal{S}_3$.*

We faithfully express $(x_1 \vee x_2 \vee x_3)$ from M using a formula derived from a gadget in [11]. This gadget expresses $(x_1 \vee x_2 \vee x_3)$ in terms of "Protected OR", which corresponds to our relation M.

$$\begin{aligned}
(x_1 \vee x_2 \vee x_3) &= \exists y_1 \ldots y_5\ (x_1 \vee \bar{y}_1) \wedge (x_2 \vee \bar{y}_2) \wedge (x_3 \vee \bar{y}_3) \wedge (x_3 \vee \bar{y}_4) \\
&\quad \wedge M(y_1, y_5, y_3) \wedge M(y_2, \bar{y}_5, y_4)
\end{aligned} \tag{1}$$

The table below shows that the conditions for faithful expressibility are satisfied.

x_1, x_2, x_3	$y_1 \ldots y_5$			
111	00011 00111 00110 00100 01100 01101	01001 11001 11000	10000	10010 10011
110		01001 11001 11000	10000	
100			10000	
101	00011 00111 00110 00100		10000	10010 10011
001	00011 00111 00110 00100			
011	00011 00111 00110 00100 01100 01101	01001		
010		01001		

From the relation $(x_1 \vee x_2 \vee x_3)$ we derive the other 3-clauses by reverse resolution, for instance

$$(\bar{x}_1 \vee x_2 \vee x_3) = \exists y\ (\bar{x}_1 \vee \bar{y}) \wedge (y \vee x_2 \vee x_3)$$

□

To complete the proof of the Faithful Expressibility Theorem, we show that an arbitrary relation can be expressed faithfully from $\mathcal{S}_3$.

Lemma 7 *Let $R \subseteq \{0,1\}^k$ be any relation of arity $k \geq 1$. R is faithfully expressible from $\mathcal{S}_3$.*

PROOF: If $k \leq 3$ then R can be expressed as a formula in CNF$(\mathcal{S}_3)$ with constants, without introducing witness variables. This kind of expression is always faithful.

If $k \geq 4$ then R can be expressed as a formula in CNF$(\mathcal{S}_k)$, without witnesses (i.e. faithfully). We will show that every k-clause can be expressed faithfully from $\mathcal{S}_{k-1}$. Then, by induction, it can be expressed faithfully from $\mathcal{S}_3$. For simplicity we express a k-clause corresponding to the relation D_0. The remaining relations are expressed equivalently. We express D_0 in a way that is standard in other complexity reductions, and turns out to be faithful:

$$(x_1 \vee x_2 \vee \cdots \vee x_k) = \exists y\ (x_1 \vee x_2 \vee y) \wedge (\bar{y} \vee x_3 \vee \cdots \vee x_k).$$

This is the reverse operation of resolution. For any satisfying assignment for $\mathbf{x}$, its witness space is either $\{0\}$, $\{1\}$ or $\{0,1\}$, so in all cases it is connected. Furthermore, the only way two neighboring satisfying assignments for x can have no common witness is if one of them has witness set $\{0\}$, and the other one has witness set $\{1\}$. This implies that the first one has $(x_3,\ldots,x_k) = (0,\ldots,0)$, and the other one has $(x_1,x_2) = (0,0)$, thus they differ in the assignments of at least two variables: one from $\{x_1,x_2\}$ and one from $\{x_3,\ldots,x_k\}$. In that case they cannot be neighboring assignments. Therefore all requirements of faithful expressibility are satisfied. $\Box$

4.3 Hardness Results for 3-CNF formulas

Lemma 6, together with Corollary 3 allows us to focus on 3-CNF formulas to prove the hard side of our dichotomies. We show that 3-CNF formulas can have exponential diameter, by inductively constructing a path of length at least $2^{\frac{n}{2}}$ on n variables and then identifying it with the solution space of a 3-CNF formula with $O(n^2)$ clauses. By Lemma 6 and Corollary 3, this implies the diameter dichotomy (Theorem 5).

Lemma 8 *For n even, there is a* 3-CNF *formula φ_n with n variables and $O(n^2)$ clauses, such that $G(\varphi_n)$ is a path of length greater than $2^{\frac{n}{2}}$.*

PROOF: The construction is in two steps: we first exhibit an induced subgraph G_n of the n dimensional hypercube with large diameter. We then construct a 3-CNF formula φ_n so that $G_n = G(\varphi_n)$.

The graph G_n is a path of length $2^{\frac{n}{2}}$. We construct it using induction. For $n = 2$, we take $V(G_2) = \{(0,0),(0,1),(1,1)\}$ which has diameter 2. Assume that we have constructed G_{n-2} with $2^{\frac{n-2}{2}}$ vertices, and with distinguished vertices $\mathbf{s_{n-2}}, \mathbf{t_{n-2}}$ such that the shortest path from $\mathbf{s}$ to $\mathbf{t}$ in G_{n-2} has length $2^{\frac{n-2}{2}}$. We now describe the set $V(G_n)$. For each vertex $\mathbf{v} \in V(G_{n-2})$, $V(G_n)$ contains two vertices $(\mathbf{v},0,0)$ and $(\mathbf{v},1,1)$. Note that the subgraph induced by these vertices alone consists of two disconnected copies of G_{n-2}. To connect these two components, we add the vertex $\mathbf{m} = (\mathbf{t},0,1)$ (which is connected to $(\mathbf{t},0,0)$ and $(\mathbf{t},1,1)$ in the induced subgraph). Note that the resulting graph G_n is connected, but any path from $(\mathbf{u},0,0)$ to $(\mathbf{v},1,1)$ must pass through $\mathbf{m}$. Further note that by induction, the graph G_n is also a path. The vertices $\mathbf{s_n} = (\mathbf{s_{n-2}},0,0)$ and $\mathbf{t_n} = (\mathbf{s_{n-2}},1,1)$ are diametrically opposite ends of this path. The path length is at least $2 \cdot 2^{\frac{n-2}{2}} + 2 > 2^{\frac{n}{2}}$. Also $\mathbf{s_2} = (0,0)$, $\mathbf{s_n} = (\mathbf{s_{n-2}},0,0)$, $\mathbf{t_n} = (\mathbf{s_{n-2}},1,1)$ and hence $\mathbf{s_n} = (0,\ldots,0), \mathbf{t_n} = (0,\ldots,0,1,1)$.

We construct a sequence of 3-CNF formulas $\varphi_n(x_1,\ldots,x_n)$ so that $G_n = G(\varphi_n)$. Let $\varphi_2(x_1,x_2) = \bar{x}_1 \vee x_2$. Assume we have $\varphi_{n-2}(x_1,\ldots,x_{n-2})$. We add two variables x_{n-1} and x_n and the clauses

$$\varphi_{n-2}(x_1,\ldots,x_{n-2}),\ \ \bar{x}_{n-1} \wedge x_n$$

$$x_{n-1} \vee \bar{x}_n \vee \bar{x}_i \qquad \text{for } i \leq n-4 \tag{2}$$

$$x_{n-1} \vee \bar{x}_n \vee x_i \qquad \text{for } i = n-3, n-2 \tag{3}$$

Note that a clause in 2 is just the implication $(\bar{x}_{n-1} \wedge x_n) \rightarrow \bar{x}_i$. Thus clauses 2, 3 enforce the condition that $x_{n-1} = 0, x_n = 1$ implies that $(x_1,\ldots,x_{n-2}) = \mathbf{t_{n-2}} = (0,\ldots,0,1,1)$. $\Box$

The proof that CONN($\mathcal{S}_3$) and ST-CONN($\mathcal{S}_3$) are PSPACE-complete is fairly intricate, and is via a direct reduction from the computation of a polynomial-space Turing machine. The result for ST-CONN can also be proved using results of Hearne and Demaine on Non-deterministic Constraint Logic [11]. It does not appear that completeness for CONN follows from their results.

Lemma 9 ST-CONN($\mathcal{S}_3$) *and* CONN($\mathcal{S}_3$) *are* PSPACE*-complete.*

PROOF: Given a CNF($\mathcal{S}_3$) formula φ and solutions s, t we can check if they are connected in $G(\varphi)$ with polynomial amount of space. Similarly for CONN($\mathcal{S}_3$), we can check for all pairs of assignments whether they are satisfying and connected in $G(\varphi)$ with polynomial amount of space, so both problems are in PSPACE.

Next we show that CONN($\mathcal{S}_3$) and ST-CONN($\mathcal{S}_3$) are PSPACE-hard. Let A be a language decided by a deterministic Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ in space n^k for some constant k. We give a polynomial time reduction from A to ST-CONN($\mathcal{S}_3$) and CONN($\mathcal{S}_3$).

The reduction maps a string w (with $|w| = n$) to a 3-CNF formula φ and two satisfying assignments for the formula, which are connected in $G(\varphi)$ if and only if M accepts w. Furthermore, all satisfying assignments of φ are connected to one of these two assignments, so that $G(\varphi)$ is connected if and only if M accepts w.

Before we show how to construct φ, we modify M in several ways:

1. We add a clock that counts from 0 to $n^k \times |Q| \times |\Gamma|^{n^k} = 2^{O(n^{k+1})}$, which is the total number of possible distinct configurations of M. It uses a separate tape of length $O(n^{k+1})$ with the alphabet $\{0, 1\}$. Before a transition happens, control is passed on to the clock, its counter is incremented, and finally the transition is completed.

2. We define a token accepting configuration. Whenever q_{accept} is reached, the clock is stopped and set to zero, the original tape is erased and the head is placed in the initial position.

3. Whenever q_{reject} is reached the machine goes into its initial configuration. First w is written back on the input tape. This step requires adding n states to the machine in order to write the n letters of w. This increases the number of states of M' to $O(n)$. Next, the rest of the tape is erased, the clock is set to zero, the head is placed in the initial position, and the state is set to q_0.

4. Whenever the clock overflows, the machine goes into q_{reject}.

The new machine M' runs forever if w is not in A and accepts if w is in A. It also has the property that every configuration leads either to the accepting configuration or to the initial configuration with input w. Therefore the space of configurations is connected if and only if $w \in A$. Let's denote by Q' the states of M' and by δ' its transitions. As mentioned earlier, $|Q'| = O(n)$, and M' runs on two tapes, one of size $N = n^k$, and the other (for the clock) of size $N_c = O(n^{k+1})$. The alphabet of M' on one tape is Γ, and on the other $\{0, 1\}$. For simplicity we can also assume that at each transition the machine uses only one of the two tapes.

Next, we construct a CNF-formula ψ whose solutions are the configurations of M'. However, the space of solutions of ψ is disconnected.

For each $i \in [N]$ and $a \in \Gamma$, we have a variable $x(i, a)$. If $x(i, a) = 1$, this means that the i^{th} tape cell contains symbol a. For every $i \in [N]$ there is a variable $y(i)$ which is 1 if the head is at position i. For every $q \in Q'$, there is a variable $z(q)$ which is 1 if the current state is q. Similarly for every $j \in [N_c]$ and $a \in \{0, 1\}$ we have variables $x_c(j, a)$ and a variable $y_c(j)$ which is 1 if the head of the clock tape is at position j.

We enforce the following conditions:

1. Every cell contains some symbol:

$$\psi_1 = \bigwedge_{i \in [N]} \left(\vee_{a \in \Gamma} \; x(i, a)\right) \bigwedge_{j \in [N_c]} \left(\vee_{a \in \{0,1\}} \; x_c(j, a)\right).$$

2. No cell contains two symbols:

$$\psi_2 = \bigwedge_{i\in[N]} \bigwedge_{a\neq a'\in\Gamma} \left(\overline{x(i,a)} \vee \overline{x(i,a')}\right) \bigwedge_{j\in[N_c]} \left(\overline{x_c(j,0)} \vee \overline{x_c(j,1)}\right).$$

3. The head is in some position, and in some state:

$$\psi_3 = \left(\vee_{i\in[N]}\ y(i)\right) \bigwedge \left(\vee_{j\in[N]}\ y_c(j)\right) \bigwedge \left(\vee_{q\in Q_1}\ z(q)\right).$$

4. The head is in a unique position, and in a unique state:

$$\psi_4 = \bigwedge_{i\neq i'\in[N]} \left(\overline{y(i)} \vee \overline{y(i')}\right) \bigwedge_{j\neq j'\in[N_c]} \left(\overline{y_c(j)} \vee \overline{y_c(j')}\right) \bigwedge_{q\neq q'\in Q'} \left(\overline{z(q)} \vee \overline{z(q')}\right).$$

Solutions of $\psi = \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4$ are in 1-1 correspondence with configurations of M'. Furthermore, the assignments corresponding to any two distinct configurations differer in at least two variables.

Next, to connect the solution space along valid transitions of M', we relax conditions 2 and 4 by introducing new transition variables, which allow the head to have two states or a cell to have two symbols at the same time. This allows us to go from one configuration to the next.

Consider a transition $\delta(q,a) = (q', b, R)$, which operates on the first tape, for example. Fix the position of the head of the first tape to be i, and the symbol in position $i+1$ to be c. The variables that are changed by the transition are: $x(i,a)$, $y(i)$, $z(q)$, $x(i,b)$, $y(i+1)$, $z(q')$. Before the transition the first three are set to 1, the second three are set to 0, and after the transition they are all flipped. Corresponding to this transition (which is specified by i, q, a, and c) we introduce a transition variable $t(i,q,a,c)$. We now relax conditions 2 and 4 as follows:

- Replace $\left(\overline{x(i,a)} \vee \overline{x(i,b)}\right)$ by $\left(\overline{x(i,a)} \vee \overline{x(i,b)} \vee t(i,q,a,c)\right)$.
- Replace $\left(\overline{y(i)} \vee \overline{y(i+1)}\right)$ by $\left(\overline{y(i)} \vee \overline{y(i+1)} \vee t(i,q,a,c)\right)$.
- Replace $\left(\overline{z(q)} \vee \overline{z(q')}\right)$ by $\left(\overline{z(q)} \vee \overline{z(q')} \vee t(i,q,a,c)\right)$.

This is done for every value of q, a, i and c (and also for transitions acting on the clock tape). We add the transition variables to the corresponding clauses so that for example the clause $\left(\overline{x(i,a)} \vee \overline{x(i,b)}\right)$ could potentially become very long, such as: $\left(\overline{x(i,a)} \vee \overline{x(i,b)} \vee t(i,q_1,a,c_1) \vee t(i,q_2,a,c_2) \vee \dots\right)$. However, the total number of transition variables is only polynomial in n. We also add a constraint for every pair of transition variables $t(i,q,a,c)$, $t(i',q',a',c')$, saying they cannot be 1 simultaneously: $(\overline{t(i,q,a,c)} \vee \overline{t(i',q',a',c')})$. This ensures that only one transition can be happening at any time. The effect of adding the transition variables to the clauses of ψ_2 and ψ_4 is that by setting $t(i,q,a,c)$ to 1, we can simultaneously set $x(i,a)$ and $x(i,b)$ to 1, and so on. This gives a path from the initial configuration to the final configuration as follows: Set $t(i,q,a,c)=1$, set $x(i,b)=1, y(i+1)=1, z(q')=1, x(i,a)=0, y(i)=0, z(q)=0$, then set $t(i,q,a,c)=0$. Thus consecutive configurations are now connected. To avoid connecting to other configurations, we also add an expression to ensure that these are the only assignments the 6 variables can take when $t(i,q,a,c)=1$:

$$\begin{aligned}\psi_{i,q,a,c} \quad = \quad & \overline{t(i,q,a,c)} \vee ((x(i,a), y(i), z(q), x(i,b)), y(i+1), z(q')) \in \\ & \qquad \{111000, 111100, 111110, 111111, 011111, 001111, 000111\}).\end{aligned}$$

This expression can of course be written in conjunctive normal form.

Call the resulting CNF formula $\varphi(\mathbf{x}, \mathbf{x_c}, \mathbf{y}, \mathbf{y_c}, \mathbf{z}, \mathbf{t})$. Note that $\varphi(\mathbf{x}, \mathbf{x_c}, \mathbf{y}, \mathbf{y_c}, \mathbf{z}, \mathbf{0}) = \psi(\mathbf{x}, \mathbf{x_c}, \mathbf{y}, \mathbf{y_c}, \mathbf{z})$, so a solution where all transition variables are 0 corresponds to a configuration of M'. To see that we have not introduced any shortcut between configurations that are not valid machine transitions, notice that in any solution of φ, at most a single transition variable can be 1. Therefore none of the transitional solutions belonging to different transitions can be adjacent. Furthermore, out of the solutions that have a transition variable set to 1, only the first and the last correspond to a valid configuration. Therefore none of the intermediate solutions can be adjacent to a solution with all transition variables set to 0.

The formula φ is a CNF formula where clause size is unbounded. We use the same reduction as in the proof of Lemma 7 to get a 3-CNF formula. By Lemma 5 and Corollary 3, ST-CONN and CONN for $\mathcal{S}_3$ are PSPACE-complete. □

By Lemma 6 and Corollary 3, this completes the proof of the dichotomies for CONN and ST-CONN (Theorems 3 and 4).

5 Discussion and Open Problems

In Section 2, we conjectured a trichotomy for CONN($\mathcal{S}$). We have made progress towards this conjecture; what remains is to pinpoint the complexity of CONN($\mathcal{S}$) when $\mathcal{S}$ is Horn or dual-Horn. We can extend our dichotomy theorem for st-connectivity to formulas without constants; the complexity of connectivity for formulas without constants is open. We conjecture that when $\mathcal{S}$ is not tight, one can improve the diameter bound from $2^{\Omega(\sqrt{n})}$ to $2^{\Omega(n)}$. Finally, we believe that our techniques can shed light on other connectivity-related problems, such as approximating the diameter and counting the number of components. For counting the number of components, using results of Creignou and Hermann [7], we can show that the problem is in P for affine, monotone and dual-monotone relations, and #P-complete otherwise.

References

[1] D. ACHLIOPTAS, P. BEAME, AND M. MOLLOY, *Exponential bounds for DPLL below the satisfiability threshold*, in Proc. 15^{th} ACM-SIAM Symp. Discrete Algorithms, 2004, pp. 132–133.

[2] D. ACHLIOPTAS, A. NAOR, AND Y. PERES, *Rigorous location of phase transitions in hard optimization problems*, Nature, 435 (2005), pp. 759–764.

[3] D. ACHLIOPTAS AND F. RICCI-TERSENGHI, *On the solution-space geometry of random constraint satisfaction problems*, in 38^{th} ACM Symp. Theory of Computing, 2006.

[4] E. BÖHLER, N. CREIGNOU, S. REITH, AND H. VOLLMER, *Playing with Boolean blocks, Part II: constraint satisfaction problems*, ACM SIGACT-Newsletter, 35 (2004), pp. 22–35.

[5] A. BULATOV, *A dichotomy theorem for constraints on a three-element set*, in Proc. 43^{rd} IEEE Symp. Foundations of Computer Science, 2002, pp. 649–658.

[6] N. CREIGNOU, *A dichotomy theorem for maximum generalized satisfiability problems*, Journal of Computer and System Sciences, 51 (1995), pp. 511–522.

[7] N. CREIGNOU AND M. HERMANN, *Complexity of generalized satisfiability counting problems*, Information and Computation, 125 (1996), pp. 1–12.

[8] N. CREIGNOU, S. KHANNA, AND M. SUDAN, *Complexity classification of Boolean constraint satisfaction problems*, SIAM Monographs on Disc. Math. Appl. 7, 2001.

[9] N. CREIGNOU AND B. ZANUTTINI, *A complete classification of the complexity of propositional abduction*, To appear in SIAM Journal on Computing, (2006).

[10] P. GOPALAN, P. KOLAITIS, E. MANEVA, AND C. PAPADIMITRIOU, *The connectivity of boolean satisfiability: Computational and structural dichotomies*, in 33^{rd} Intl. Colloquium on Automata, Languages and Programming, 2006, pp. 346–357.

[11] R. HEARNE AND E. DEMAINE, *The Nondeterministic Constraint Logic model of computation: Reductions and applications*, in 29^{th} Intl. Colloquium on Automata, Languages and Programming, 2002, pp. 401–413.

[12] L. JUBAN, *Dichotomy theorem for the generalized unique satisfiability problem*, in 12^{th} Intl. Symp. Fundamentals of Computation Theory, 1999, pp. 327–337.

[13] D. KAVVADIAS AND M. SIDERI, *The inverse satisfiability problem*, SIAM J. Comput., 28 (1998), pp. 152–163.

[14] S. KHANNA, M. SUDAN, L. TREVISAN, AND D. WILLIAMSON, *The approximability of constraint satisfaction problems*, SIAM J. Comput., 30(6):1863–1920, (2001).

[15] L. KIROUSIS AND P. KOLAITIS, *The complexity of minimal satisfiability problems*, Information and Computation, 187 (2003), pp. 20–39.

[16] E. MANEVA, E. MOSSEL, AND M. J. WAINWRIGHT, *A new look at survey propagation and its generalizations*, in Proc. 16^{th} ACM-SIAM Symp. Discrete Algorithms, 2005, pp. 1089–1098.

[17] M. MÉZARD, G. PARISI, AND R. ZECCHINA, *Analytic and algorithmic solution of random satisfiability problems*, Science, 297, 812 (2002).

[18] M. MÉZARD AND R. ZECCHINA, *Random k-satisfiability: from an analytic solution to an efficient algorithm*, Phys. Rev. E, 66 (2002).

[19] T. MORA, M. MÉZARD, AND R. ZECCHINA, *Clustering of solutions in the random satisfiability problem*, Phys. Rev. Lett., (2005). In press.

[20] T. SCHAEFER, *The complexity of satisfiability problems*, in Proc. 10^{th} ACM Symp. Theory of Computing, 1978, pp. 216–226.

[21] B. SELMAN, H. KAUTZ, AND B. COHEN, *Local search strategies for satisfiability testing*, in Cliques, coloring, and satisfiability : second DIMACS implementation challenge, October 1993, AMS, 1996.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/