The Connectivity of Boolean Satisfiability: Dichotomies for Formulas and Circuits

Konrad Schwerdtfeger

Institut für Theoretische Informatik, Leibniz Universität Hannover,
Appelstr. 4, 30167 Hannover, Germany
k.w.s@gmx.net

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 CSPs [3]. 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 were refined by Makino et al. [7]. Recently, we were able to establish the trichotomy [15].

Here, we consider connectivity issues of satisfiability problems defined by Boolean circuits and propositional formulas that use gates, resp. connectives, from a fixed set of Boolean functions. We obtain dichotomies for the diameter and the connectivity problems: on one side, the diameter is linear and both problems are in P, while on the other, the diameter can be exponential and the problems are PSPACE-complete.

1 Introduction

The Boolean satisfiability problem, as well as many related questions like equivalence, counting, enumeration, and numerous versions of optimization, are of great importance in both theory and applications of computer science.

Common to all these problems is that one asks questions about a Boolean relation given by some short description, e.g. a propositional formula, Boolean circuit, binary decision diagram, or Boolean neural network. For the usual formulas with the connectives $\wedge$, $\vee$ and $\neg$, several generalizations and restrictions have been considered. Most widely studied are Boolean constraint satisfactions problems (*CSP*s), that can be seen as a generalization of formulas in *CNF* (conjunctive normal form), see Definition 2. Another generalization, that we will consider here, are formulas with connectives from an arbitrary fixed set of Boolean functions B, known as *B-formulas.* This concept also applies to circuits, where the allowed gates implement the functions from B, called *B-circuits.* A further extension that allows for shorter representations, and in turn makes many problems harder, are quantifiers, which we will look at in Section 5.

E.A. Hirsch et al. (Eds.): CSR 2014, LNCS 8476, pp. 351–364, 2014.

Here we will investigate the structure of the solution space, which is of obvious relevance to these satisfiability related problems. 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 [9,8]. 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 [8]. This research was a motivation for Gopalan et al. to study connectivity properties of the solution space of Boolean CSPs [3].

While the most efficient satisfiability solvers take CNF-formulas as input, one of the most important applications of satisfiability testing is verification and optimization in Electronic Design Automation (EDA), where the instances derive mostly from digital circuit descriptions [18]. Though many such instances can easily be encoded in CNF, the original structural information, such as signal ordering, gate orientation and logic paths, is lost, or at least obscured. Since exactly this information can be very helpful for solving these instances, considerable effort has been made recently to develop satisfiability solvers that work with the circuit description directly [18], which have far superior performance in EDA applications, or to restore the circuit structure from CNF [2]. This is one major motivation for our study.

A direct application of *st*-connectivity are *reconfiguration problems*, that arise when we wish to find a step-by-step transformation between two feasible solutions of a problem 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 [4,5], and many complexity results were obtained, in some cases making use of Gopalan et al.'s results.

Since many of the satisfiability related problems are hard to solve in general (they are NP- or even PSPACE-complete), one has tried to identify easier fragments and to classify restrictions in terms of their complexity. Possibly the best known result is Schaefer's 1978 dichotomy theorem for CSPs, which states that for certain classes of allowed constraints the satisfiability of a CSP is in P, while it is NP-complete for all other classes [13]. Analogously, Gopalan et al. in 2006 classified the complexity of connectivity questions for CSPs in Schaefer's framework. In this paper, we consider the same connectivity issues as Gopalan et al., but for problems defined by Boolean circuits and propositional formulas that use gates, resp. connectives, from a fixed set of Boolean functions.

2 Propositional Formulas and Their Solution Space Connectivity

Definition 1. *An n-ary* Boolean relation *is a subset of* $\{0,1\}^n$ *($n \geq 1$). The set of solutions of a propositional formula ϕ with n variables defines in a natural way* *an n-ary Boolean relation R, where the variables are taken in lexicographic order. The* solution graph $G(\phi)$ *of ϕ is the subgraph of the n-dimensional hypercube graph induced by the vectors in R, i.e., the vertices of $G(\phi)$ are the vectors in R, and there is an edge between two vectors precisely if they differ in exactly one position.*

We use $\boldsymbol{a}, \boldsymbol{b}, \ldots$ to denote vectors of Boolean values and $\boldsymbol{x}, \boldsymbol{y}, \ldots$ to denote vectors of variables, $\boldsymbol{a} = (a_1, a_2, \ldots)$ and $\boldsymbol{x} = (x_1, x_2, \ldots)$. The Hamming distance *$|\boldsymbol{a} - \boldsymbol{b}|$ of two Boolean vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is the number of positions in which they differ. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are solutions of ϕ and lie in the same connected component of $G(\phi)$, we write $d_\phi(\boldsymbol{a}, \boldsymbol{b})$ to denote the shortest-path distance between $\boldsymbol{a}$ and $\boldsymbol{b}$. The* diameter of a connected 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 of its connected components.*

In our proofs for B-formulas and B-circuits, we will use Gopalan et al.'s results for 3-CNF-formulas, so we also need to introduce some terminology for constraint satisfaction problems.

Definition 2. *A* CNF-formula *is a Boolean formula of the form $C_1 \wedge \cdots \wedge C_m$ ($1 \leq m < \infty$), 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.*

For a finite set of Boolean relations $\mathcal{S}$, a CNF($\mathcal{S}$)-formula *(with constants) over a set of variables V is a finite conjunction $C_1 \wedge \cdots \wedge C_m$, where each C_i is a* constraint application *(*constraint *for short), i.e., an expression of the form $R(\xi_1, \ldots, \xi_k)$, with a k-ary relation $R \in \mathcal{S}$, and each ξ_j is a variable in V or one of the constants $0, 1$.*

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 $S_k = \{D_0, \ldots, D_k\}$ Thus, CNF(S_k) is the collection of k-CNF-formulas.*

Gopalan et al. studied the following two decision problems for CNF($\mathcal{S}$)-formulas:

- the *connectivity problem* Conn*($\mathcal{S}$)*: given a CNF($\mathcal{S}$)-formula ϕ, is $G(\phi)$ connected? (if ϕ is unsatisfiable, then $G(\phi)$ is considered connected)
- the *st-connectivity problem* st-Conn*($\mathcal{S}$)*: given a CNF($\mathcal{S}$)-formula ϕ and two solutions $\boldsymbol{s}$ and $\boldsymbol{t}$, is there a path from $\boldsymbol{s}$ to $\boldsymbol{t}$ in $G(\phi)$?

Lemma 1. *[3, Lemm 3.6]* st-Conn(S_3) *and* Conn(S_3) *are* PSPACE*-complete.*

Proof. st-Conn(S_3) and Conn(S_3) are in PSPACE: Given a CNF(S_3)-formula ϕ and two solutions $\boldsymbol{s}$ and $\boldsymbol{t}$, we can guess a path of length at most 2^n between them and verify that each vertex along the path is indeed a solution. Hence st-Conn(S_3) is in NPSPACE=PSPACE. For Conn(S_3), by reusing space we can check for all pairs of vectors whether they are satisfying and, if they both are, whether they are connected in $G(\phi)$.

We can not state the full proof for the PSPACE-hardness here. It consists of a direct reduction from the computation of a space-bounded Turing machine M.

The input-string w of M is mapped to a CNF(S_3)-formula and two satisfying assignments $\boldsymbol{s}$ and $\boldsymbol{t}$, corresponding to the initial and accepting configuration respectively, s.t. $\boldsymbol{s}$ and $\boldsymbol{t}$ are connected in $G(\phi)$ iff M accepts w. □

Lemma 2. *[3, Lemm 3.7] For $n \geq 2$, there is an n-ary Boolean function f with $f(1,\dots,1) = 1$ and a diameter of at least $2^{\lfloor \frac{n}{2} \rfloor}$.*

3 Circuits, Formulas, and Post's Lattice

An n-ary *Boolean function* is a function $f : \{0,1\}^n \to \{0,1\}$. Let B be a finite set of Boolean functions.

A *B-circuit* $\mathcal{C}$ with input variables $x_1,\dots,x_n$ is a directed acyclic graph, augmented as follows: Each node (here also called *gate*) with indegree 0 is labeled with an x_i or a 0-ary function from B, each node with indegree $k > 0$ is labeled with a k-ary function from B. The edges (here also called *wires*) pointing into a gate are ordered. One node is designated the output gate.

Given values $a_1,\dots,a_n \in \{0,1\}$ to $x_1,\dots,x_n$, $\mathcal{C}$ computes an n-ary function $f_{\mathcal{C}}$ as follows: A gate v labeled with a variable x_i returns a_i, a gate v labeled with a function f computes the value $f(b_1,\dots,b_k)$, where $b_1,\dots,b_k$ are the values computed by the predecessor gates of v, ordered according to the order of the wires. For a more formal definition see [17].

A *B-formula* is defined inductively: A variable x is a B-formula. If $\phi_1,\dots,\phi_m$ are B-formulas, and f is an n-ary function from B, then $f(\phi_1,\dots,\phi_n)$ is a B-formula; here, we identify the function f and the symbol representing it in a formula.

It is easy to see that the functions computable by a B-circuit, as well as the functions definable by a B-formula, are exactly those that can be obtained from B by *superposition*, together with all projections [1]. By superposition, we mean substitution (that is, composition of functions), permutation and identification of variables, and introduction of *fictive variables* (variables on which the value of the function does not depend). This class of functions is denoted by $[B]$. B is *closed* (or said to be a *clone*) if $[B] = B$. A *base* of a clone F is any set B with $[B] = F$.

Already in the early 1920s, Emil Post extensively studied Boolean functions, identified all closed classes, found a finite base for each of them, and detected their inclusion structure [11]. The closed classes form a lattice, called *Post's lattice*, depicted in Figure 1; a table of the bases can be found e.g. in [1], a modern proof e.g. in [19]. The classes are defined as follows:

- BF is the class of all Boolean functions.
- For $a \in \{0,1\}$, an n-ary Boolean function f is called *a-reproducing*, if $f(a,\dots,a) = a$; the classes R_a contain all a-reproducing functions.
- f is called *monotonic*, if $a_1 \leq b_1,\dots,a_n \leq b_n$ implies $f(a_1,\dots,a_n) \leq f(b_1,\dots,b_n)$; M is the class of all monotonic functions.
- f is called *self-dual*, if $f(x_1,\dots,x_n) = \overline{f(\overline{x_1},\dots,\overline{x_n})}$; D is the class of all self-dual functions.

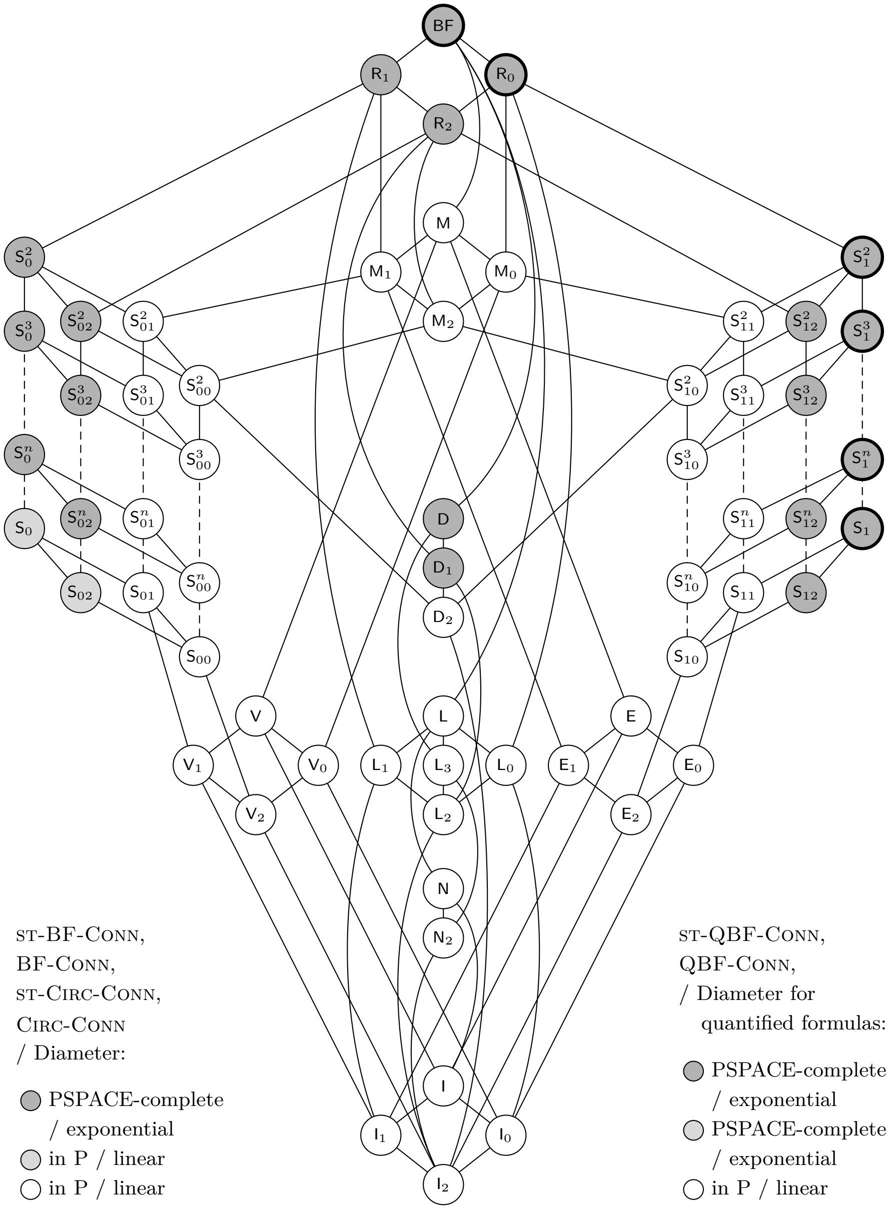

Fig. 1. Post's lattice with our results for the connectivity problems and the diameter. For comparison, the satisfiability problem (without quantifiers) is NP-complete for the bold circled classes, and in P for the other ones.

- f is called *affine*, if $f(x_1, \ldots, x_n) = x_{i_1} \oplus \cdots \oplus x_{i_m} \oplus c$ with $i_1, \ldots, i_m \in \{1, \ldots, n\}$ and $c \in \{0, 1\}$; L is the class of all affine functions.
- For $c \in \{0, 1\}$, f is called *c-separating*, if there exists an $i \in \{1, \ldots, n\}$ s.t. $a_i = c$ for all $\boldsymbol{a} \in f^{-1}(c)$; the classes S_c contain all c-separating functions.
- For $c \in \{0, 1\}$ and $k \geq 2$, f is called *c-separating of degree k*, if for all $U \subseteq f^{-1}(c)$ of size $|U| = k$ there exists an $i \in \{1, \ldots, n\}$ s.t. $a_i = c$ for all $\boldsymbol{a} \in U$; the classes S_c^k contain all c-separating functions of degree k.
- The class E contains the constant functions and all conjunctions.
- The class V contains the constant functions and all disjunctions.
- f is called a projection, if there exists an $i \in \{1, \ldots, n\}$ s.t. $f(x_1, \ldots, x_n) = x_i$; The class I contains the constant functions and all projections.
- The class N contains the constant functions, all projections and all negations of projections.
- All other classes are defined from the above by intersection according to Post's lattice.

Not surprisingly, the complexity of problems defined by B-formulas and B-circuits depends on $[B]$, and the complexity of numerous problems for B-circuits and B-formulas has been classified by means of Post's lattice [12,14], starting with satisfiability: Analogously to Schaefer, Lewis in 1978 found a dichotomy for B-formulas [6]; if $[B]$ contains the function $x \wedge \overline{y}$, SAT is NP-complete, else it is in P.

While for B-circuits the complexity of every decision problem solely depends on $[B]$ (up to AC^0 isomorphism), for formulas this need not be the case, since the transformation of a B-formula into a B'-formula might require an exponential increase in the formula size even if $[B] = [B']$, as the B'-representation of some function from B may need to use some input variable more than once [10]. For example, let $h(x, y) = x \wedge \overline{y}$; then there is no shorter $\{h\}$-representation of the function $x \wedge y$ than $h(x, h(x, y))$.

4 Computational and Structural Dichotomies

Now we consider the connectivity problems for B-formulas and B-circuits:

- BF-CONN(B): Given a B-formula ϕ, is $G(\phi)$ connected?
- ST-BF-CONN(B): Given a B-formula ϕ and two solutions $\boldsymbol{s}$ and $\boldsymbol{t}$, is there a path from $\boldsymbol{s}$ to $\boldsymbol{t}$ in $G(\phi)$?

The corresponding problems for circuits are denoted CIRC-CONN(B) resp. ST-CIRC-CONN(B).

Theorem 1. *Let B be a finite set of Boolean functions.*

1. *If $B \subseteq \mathsf{M}$, $B \subseteq \mathsf{L}$, or $B \subseteq \mathsf{S}_0$, then*
   (a) ST-CIRC-CONN(B) *and* CIRC-CONN(B) *are in* P*,*
   (b) ST-BF-CONN(B) *and* BF-CONN(B) *are in* P*,*
   (c) *the diameter of every function $f \in [B]$ is linear in the number of variables of f.*

2. *Otherwise,*
   (a) st-Circ-Conn(B) *and* Circ-Conn(B) *are* PSPACE*-complete,*
   (b) st-BF-Conn(B) *and* BF-Conn(B) *are* PSPACE*-complete,*
   (c) *there are functions* $f \in [B]$ *such that their diameter is exponential in the number of variables of* f*.*

The proof follows from the Lemmas in the next subsections. By the following Proposition, we can relate the complexity of B-formulas and B-circuits.

Proposition 1. *Every B-formula can be transformed into an equivalent B-circuit in polynomial time.*

Proof. A B-formula already is a suitable encoding for a special B-circuit with outdegree of at most one. □

4.1 The Easy Side of the Dichotomy

Lemma 3. *If $B \subseteq \mathsf{M}$, the solution graph of any n-ary function $f \in [B]$ is connected, and $d_f(\boldsymbol{a}, \boldsymbol{b}) = |\boldsymbol{a} - \boldsymbol{b}| \leq n$ for any two solutions $\boldsymbol{a}$ and $\boldsymbol{b}$.*

Proof. The table of all closed classes of Boolean functions shows that f is monotonic in this case. Thus, either $f = 0$, or $(1, \ldots, 1)$ must be a solution, and every other solution $\boldsymbol{a}$ is connected to $(1, \ldots, 1)$ in $G(\phi)$ since $(1, \ldots, 1)$ can be reached by flipping the variables assigned 0 in $\boldsymbol{a}$ one at a time to 1. Further, if $\boldsymbol{a}$ and $\boldsymbol{b}$ are solutions, $\boldsymbol{b}$ can be reached from $\boldsymbol{a}$ in $|\boldsymbol{a} - \boldsymbol{b}|$ steps by first flipping all variables that are assigned 0 in $\boldsymbol{a}$ and 1 in $\boldsymbol{b}$, and then flipping all variables that are assigned 1 in $\boldsymbol{a}$ and 0 in $\boldsymbol{b}$. □

Lemma 4. *If $B \subseteq \mathsf{S}_0$, the solution graph of any function $f \in [B]$ is connected, and $d_f(\boldsymbol{a}, \boldsymbol{b}) \leq |\boldsymbol{a} - \boldsymbol{b}| + 2$ for any two solutions $\boldsymbol{a}$ and $\boldsymbol{b}$.*

Proof. Since f is 0-separating, there is an i such that $a_i = 0$ for every vector $\boldsymbol{a}$ with $f(\boldsymbol{a}) = 0$, thus every $\boldsymbol{b}$ with $b_i = 1$ is a solution. It follows that every solution $\boldsymbol{t}$ can be reached from any solution $\boldsymbol{s}$ in at most $|\boldsymbol{s} - \boldsymbol{t}| + 2$ steps by first flipping the i-th variable from 0 to 1 if necessary, then flipping all other variables in which $\boldsymbol{s}$ and $\boldsymbol{t}$ differ, and finally flipping back the i-th variable if necessary. □

Lemma 5. *If $B \subseteq \mathsf{L}$,*

1. st-Circ-Conn(B) *and* Circ-Conn(B) *are in P,*
2. st-BF-Conn(B) *and* BF-Conn(B) *are in P,*
3. *for any function $f \in [B]$, $d_f(\boldsymbol{a}, \boldsymbol{b}) = |\boldsymbol{a} - \boldsymbol{b}|$ for any two solutions $\boldsymbol{a}$ and $\boldsymbol{b}$ that lie in the same connected component of $G(\phi)$.*

Proof. Since every function $f \in \mathsf{L}$ is linear, $f(x_1, \ldots, x_n) = x_{i_1} \oplus \ldots \oplus x_{i_m} \oplus c$, and any two solutions $\boldsymbol{s}$ and $\boldsymbol{t}$ are connected iff they differ only in fictional variables: If $\boldsymbol{s}$ and $\boldsymbol{t}$ differ in at least one non-fictional variable (i.e., an $x_i \in \{x_{i_1}, \ldots, x_{i_m}\}$), to reach $\mathbf{t}$ from $\mathbf{s}$, x_i must be flipped eventually, but for every solution $\boldsymbol{a}$, any vector $\boldsymbol{b}$ that differs from $\boldsymbol{a}$ in exactly one non-fictional variable is no solution. If $\boldsymbol{s}$ and $\boldsymbol{t}$ differ only in fictional variables, $\boldsymbol{t}$ can be reached from $\boldsymbol{s}$ in $|\boldsymbol{s}-\boldsymbol{t}|$ steps by flipping one by one the variables in which they differ.

Since $\{x \oplus y, 1\}$ is a base of L (see Fig. 1 int [1]), every B-circuit $\mathcal{C}$ can be transformed in polynomial time into an equivalent $\{x \oplus y, 1\}$-circuit $\mathcal{C}'$ by replacing each gate of $\mathcal{C}'$ with an equivalent $\{x \oplus y, 1\}$-circuit. Now one can decide in polynomial time whether a variable x_i is fictional by checking for $\mathcal{C}'$ whether the number of "backward paths" from the output gate to gates labeled with x_i is odd, so st-Circ-Conn(B) is in P.

$G(\mathcal{C})$ is connected iff at most one variable is non-fictional, thus Circ-Conn(B) is in P.

By Proposition 1, st-BF-Conn(B) and BF-Conn(B) are in P also. □

This completes the proof of the easy side of the dichotomy.

4.2 The Hard Side of the Dichotomy

Proposition 2. st-Circ-Conn(B) *and* Circ-Conn(B), *as well as* st-BF-Conn(B) *and* BF-Conn(B), *are in* PSPACE *for any finite set B of Boolean functions.*

Proof. This follows as in Lemma 1. □

Proposition 3. *For 1-reproducing 3-CNF-formulas, the problems* st-Conn *and* Conn *are* PSPACE-*complete.*

Proof. We chose the variables in the proof of Lemma 1 such that the accepting configuration of the Turing machine corresponds to the $(1, \ldots, 1)$ vector. □

An inspection of Post's lattice shows that if $B \not\subseteq \mathsf{M}$, $B \not\subseteq \mathsf{L}$, and $B \not\subseteq \mathsf{S}_0$, then $[B] \supseteq \mathsf{S}_{12}$, $[B] \supseteq \mathsf{D}_1$, or $[B] \supseteq \mathsf{S}_{02}^k, \forall k \geq 2$, so we have to prove PSPACE-completeness and show the existence of B-formulas with an exponential diameter in these cases.

In the proofs, we will use the following abbreviations: If we have the n variables $x_1, \ldots, x_n$, we write $\boldsymbol{x}$ for $x_1 \wedge \cdots \wedge x_n$ and $\overline{\boldsymbol{x}}$ for $\overline{x}_1 \wedge \cdots \wedge \overline{x}_n$. Also, we write $(\boldsymbol{x} = c_1 \cdots c_n)$ for $x_1 \leftrightarrow c_1 \wedge \cdots \wedge x_n \leftrightarrow c_n$, where $c_1, \ldots, c_n \in \{0,1\}$ are constants; e.g., we write $(\boldsymbol{x} = 101)$ for $x_1 \wedge \overline{x}_2 \wedge x_3$. Further, we use $\boldsymbol{x} \in \{\boldsymbol{a}, \boldsymbol{b}, \ldots\}$ for $(\boldsymbol{x} = \boldsymbol{a}) \vee (\boldsymbol{x} = \boldsymbol{b}) \vee \ldots$. Finally, if we have two vectors of Boolean values $\boldsymbol{a}$ and $\boldsymbol{b}$ of length n and m resp., we write $\boldsymbol{a} \cdot \boldsymbol{b}$ for their concatenation $(a_1, \ldots, a_n, b_1, \ldots b_m)$.

Lemma 6. *If* $[B] \supseteq \mathsf{S}_{12}$,

1. st-BF-Conn(B) *and* BF-Conn(B) *are* PSPACE-*complete,*
2. st-Circ-Conn(B) *and* Circ-Conn(B) *are* PSPACE-*complete,*
3. *for $n \geq 3$, there is an n-ary function $f \in [B]$ with diameter of at least* $2^{\lfloor \frac{n-1}{2} \rfloor}$.

Proof. 1. We reduce the problems for 1-reproducing 3-CNF-formulas to the ones for B-formulas: We map a 1-reproducing 3-CNF-formula ϕ and two solutions $\boldsymbol{s}$ and $\boldsymbol{t}$ of ϕ to a B-formula ϕ' and two solutions $\boldsymbol{s'}$ and $\boldsymbol{t'}$ of ϕ' such that $\boldsymbol{s'}$ and $\boldsymbol{t'}$ are connected in $G(\phi')$ iff $\boldsymbol{s}$ and $\boldsymbol{t}$ are connected in $G(\phi)$, and such that $G(\phi')$ is connected iff $G(\phi)$ is connected.

First for any 1-reproducing formula ψ, we define a connectivity-equivalent formula $T_\psi \in \mathsf{S}_{12}$ using the standard connectives, then we show how to transform ϕ into the B-formula ϕ' that will be equivalent to T_ϕ.

Let ψ be a 1-reproducing formula over the variables $x_1, \ldots, x_n$. We define the formula T_ψ over the $n+1$ variables $x_1, \ldots, x_n$ and y as

$$T_\psi = \psi \wedge y,$$

where y is a new variable. All solutions $\boldsymbol{a}$ of $T_\psi(\boldsymbol{x}, y)$ have $a_{n+1} = 1$, so T_ψ is 1-seperating and 0-reproducing. Moreover, T_ψ is still 1-reproducing, and thus in S_{12}. For any two solutions $\boldsymbol{s}$ and $\boldsymbol{t}$ of $\psi(\boldsymbol{x})$, $\boldsymbol{s'} = \boldsymbol{s} \cdot 1$ and $\boldsymbol{t'} = \boldsymbol{t} \cdot 1$ are solutions of $T_\psi(\boldsymbol{x}, y)$, and it is easy to see that they are connected in $G(T_\psi)$ iff $\boldsymbol{s}$ and $\boldsymbol{t}$ are connected in $G(\psi)$, and that $G(T_\psi)$ is connected iff $G(\psi)$ is connected.

Now we know that for any 1-reproducing 3-CNF-formula ϕ, T_ϕ can be expressed as a B-formula ϕ' since $T_\phi \in \mathsf{S}_{12}$. However, the transformation could lead to an exponential increase in the formula size (see Section 3), so we have to show how to construct ϕ' in polynomial time. We do this by parenthesizing the conjunctions of ϕ such that we get a tree of $\wedge$'s of depth logarithmic in the size of ϕ, and then replacing each clause C_i with some B-formula ξ_{C_i}, and each expression $\phi_1 \wedge \phi_2$ with a B-formula $\xi_\wedge(\phi_1, \phi_2)$, s.t. the resulting formula is equivalent to T_ϕ. This can increase the formula size by only a polynomial in the original size even if $\xi_\wedge$ uses some input variable more than once. This is a standard-technique for such proofs in Post's framework, see e.g. [1]. Here we easily see that we can simply replace each clause C_i of ϕ with some B-formula equivalent to T_{C_i} and each $\wedge$ with a B-formula equivalent to $T_\wedge$ since $(\psi_1 \wedge y) \wedge (\psi_2 \wedge y) \wedge y \equiv \psi_1 \wedge \psi_2 \wedge y$, but in the next proofs this will not be obvious, so we formalize the procedure.

Let $\phi = C_1 \wedge \cdots \wedge C_n$ be a 1-reproducing 3-CNF-formula. Since ϕ is 1-reproducing, every clause C_i of ϕ is itself 1-reproducing, and we can express T_{C_i} through a B-formula $T^*_{C_i}$. Also, we can express $T_\wedge(x_1, x_2) = x_1 \wedge x_2 \wedge y$ through a B-formula $T^*_\wedge$ since $\wedge$ is 1-reproducing. Now let $\phi' = \textsc{Tr}\left(T^*_{C_1}, \ldots, T^*_{C_n}\right)$, where $\textsc{Tr}$ is the following recursive algorithm that takes a list of formulas as input,

Algorithm $\textsc{Tr}(\psi_1, \ldots, \psi_m)$

1. if $m = 1$ return ψ_1
2. else if m is even, return
   $\textsc{Tr}(T^*_\wedge[x_1/\psi_1, x_2/\psi_2], T^*_\wedge[x_1/\psi_3, x_2/\psi_4], \ldots, T^*_\wedge[x_1/\psi_{m-1}, x_2/\psi_m])$
3. else return
   $\textsc{Tr}(T^*_\wedge[x_1/\psi_1, x_2/\psi_2], T^*_\wedge[x_1/\psi_3, x_2/\psi_4], \ldots, T^*_\wedge[x_1/\psi_{m-2}, x_2/\psi_{m-1}], \psi_m)$.

Here $\psi[x_i/\xi]$ denotes the formula obtained by substituting the formula ξ for the variable x_i in the formula ψ. Note that in every T^*_ψ we have the *same* variable y.

Since the recursion terminates after a number of steps logarithmic in the number of clauses of ϕ, and every step increases the total formula size by only a constant factor, the algorithm runs in polynomial time. We show $\phi' = T_\phi$ by induction. The basis is clear. Since $T_\psi \equiv T^*_\psi$, it suffices to show that $T_\wedge[x_1/T_{\psi_1}, x_2/T_{\psi_2}] \equiv T_{\psi_1 \wedge \psi_2}$:

$$T_\wedge[x_1/T_{\psi_1}, x_2/T_{\psi_2}] = T_{\psi_1} \wedge T_{\psi_2} \wedge y = (\psi_1 \wedge y) \wedge (\psi_2 \wedge y) \wedge y \equiv \psi_1 \wedge \psi_2 \wedge y = T_{\psi_1 \wedge \psi_2}.$$

2. This follows from 1. by Proposition 1.

3. By Lemma 2 there is an 1-reproducing $(n-1)$-ary function f with diameter of at least $2^{\lfloor \frac{n-1}{2} \rfloor}$. Let f be represented by a formula ϕ; then, T_ϕ represents an n-ary function of the same diameter in S_{12}. □

Lemma 7. *If* $[B] \supseteq \mathsf{D}_1$,

1. st-BF-Conn(B) *and* BF-Conn(B) *are* PSPACE-*complete,*
2. st-Circ-Conn(B) *and* Circ-Conn(B) *are* PSPACE-*complete,*
3. *for* $n \geq 5$*, there is an* n*-ary function* $f \in [B]$ *with diameter of at least* $2^{\lfloor \frac{n-3}{2} \rfloor}$.

Proof. 1. This proof is similar to the previous one, but the construction is more intricate; for every 1-reproducing 3-CNF formula we have to construct a self-dual function s.t. the connectivity is retained. For clarity, we do the construction in two steps.

For a 1-reproducing formula ψ over the n variables $x_1, \ldots, x_n$, we construct a formula $T^\sim_\psi \in \mathsf{D}_1$ with three new variables $(y_1, y_2, y_3) = \boldsymbol{y}$,

$$T^\sim_\psi = (\psi(\boldsymbol{x}) \wedge \boldsymbol{y}) \vee \left(\overline{\psi(\overline{\boldsymbol{x}})} \wedge \overline{\boldsymbol{y}}\right) \vee \boldsymbol{y} \in \{100, 010, 001\}.$$

Observe that $T^\sim_\psi(\boldsymbol{x}, \boldsymbol{y})$ is self-dual: for any solution ending with 111, the inverse vector (that ends with 000) is no solution; all vectors ending with 100, 010, or 001 are solutions and their inverses are no solutions. Also, $T^\sim_\psi$ is still 1-reproducing, and it is 0-reproducing since $\overline{\psi(\overline{0 \cdots 0})} \equiv \overline{\psi(1 \cdots 1)} \equiv 0$.

Further, for any two solutions $\boldsymbol{s}$ and $\boldsymbol{t}$ of $\psi(\boldsymbol{x})$, $\boldsymbol{s}' = \boldsymbol{s} \cdot 111$ and $\boldsymbol{t}' = \boldsymbol{t} \cdot 111$ are solutions of $T^\sim_\psi(\boldsymbol{x}, \boldsymbol{y})$ and are connected in $G(T^\sim_\psi)$ iff $\boldsymbol{s}$ and $\boldsymbol{t}$ are connected in $G(\psi)$: Every solution $\boldsymbol{a}$ of ψ corresponds to a solution $\boldsymbol{a} \cdot 111$ of $T^\sim_\psi$, and the connectivity does not change by padding the vectors with 111, and since there are no solutions of $T^\sim_\psi$ ending with 110, 101, or 011, every other solution of $T^\sim_\psi$ differs in at least two variables from the solutions $\boldsymbol{a} \cdot 111$ that correspond to solutions of ψ.

Note that exactly one connected component is added in $G(T^\sim_\psi)$ to the components corresponding to those of $G(\psi)$: It consists of all vectors ending with 000, 100, 010, or 001 (any two vectors ending with 000 are connected e.g. via those ending with 001). It follows that $G(T^\sim_\psi)$ is always unconnected. To fix this, we modify $T^{\sim}_{\psi}$ to a function T_ψ by adding $1\cdots 1\cdot 110$ as a solution, thereby connecting $1\cdots 1\cdot 111$ (which is always a solution because $T^{\sim}_{\psi}$ is 1-reproducing) with $1\cdots 1\cdot 100$, and thereby with the additional component of T_ψ. To keep the function self-dual, we must in turn remove $0\cdots 0\cdot 001$, which does not alter the connectivity. Formally,

$$\begin{aligned} T_\psi &= \left(T^{\sim}_{\psi} \vee (\boldsymbol{x} \wedge (\boldsymbol{y} = 110))\right) \wedge \neg(\overline{\boldsymbol{x}} \wedge (\boldsymbol{y} = 001)) \\ &= (\psi(\boldsymbol{x}) \wedge \boldsymbol{y}) \vee \left(\overline{\psi(\overline{\boldsymbol{x}})} \wedge \overline{\boldsymbol{y}}\right) \\ &\quad \vee (\boldsymbol{y} \in \{100, 010, 001\} \wedge \neg(\overline{\boldsymbol{x}} \wedge (\boldsymbol{y} = 001))) \vee (\boldsymbol{x} \wedge (\boldsymbol{y} = 110)). \end{aligned} \tag{1}$$

Now $G(T_\psi)$ is connected iff $G(\psi)$ is connected.

Next again we use the algorithm Tr from the previous proof to transform any 1-reproducing 3-CNF-formula ϕ into a B-formula ϕ' equivalent to T_ϕ, but with the definition (1) of T. Again, we have to show $T_\wedge[x_1/T_{\psi_1}, x_2/T_{\psi_2}] \equiv T_{\psi_1 \wedge \psi_2}$. Here,

$$\begin{aligned} T_\wedge[x_1/T_{\psi_1}, x_2/T_{\psi_2}] = {} & (T_{\psi_1} \wedge T_{\psi_2} \wedge \boldsymbol{y}) \vee \left(\overline{\overline{T_{\psi_1}} \wedge \overline{T_{\psi_2}}} \wedge \overline{\boldsymbol{y}}\right) \\ & \vee \left(\boldsymbol{y} \in \{100, 010, 001\} \wedge \neg\left(\overline{T_{\psi_1}} \wedge \overline{T_{\psi_2}} \wedge (\boldsymbol{y} = 001)\right)\right) \\ & \vee (T_{\psi_1} \wedge T_{\psi_2} \wedge (\boldsymbol{y} = 110)). \end{aligned}$$

We consider the parts of the formula in turn: For any formula ξ we have $T_\xi(\boldsymbol{x}_\xi) \wedge \boldsymbol{y} \equiv \xi(\boldsymbol{x}_\xi) \wedge \boldsymbol{y}$ and $T_\xi(\boldsymbol{x}_\xi) \wedge \overline{\boldsymbol{y}} \equiv \overline{\psi(\overline{\boldsymbol{x}_\xi})} \wedge \overline{\boldsymbol{y}}$, where $\boldsymbol{x}_\xi$ denotes the variables of ξ. Using $\overline{\overline{T_{\psi_1}(\boldsymbol{x}_{\psi_1})} \wedge \overline{T_{\psi_2}(\boldsymbol{x}_{\psi_2})}} \wedge \overline{\boldsymbol{y}} = (T_{\psi_1}(\boldsymbol{x}_{\psi_1}) \vee T_{\psi_2}(\boldsymbol{x}_{\psi_2})) \wedge \overline{\boldsymbol{y}}$, the first line becomes

$$(\psi_1(\boldsymbol{x}_{\psi_1}) \wedge \psi_2(\boldsymbol{x}_{\psi_2}) \wedge \boldsymbol{y}) \vee \left(\left(\overline{\psi_1(\overline{\boldsymbol{x}_{\psi_1}}) \wedge \psi_2(\overline{\boldsymbol{x}_{\psi_2}})}\right) \wedge \overline{\boldsymbol{y}}\right).$$

For the second line, we observe $\overline{T_\psi(\boldsymbol{x}_\psi)} \equiv \left(\overline{\psi(\boldsymbol{x}_\psi)} \vee \neg(\boldsymbol{y})\right) \wedge (\psi(\overline{\boldsymbol{x}}_{\boldsymbol{\psi}}) \vee \neg(\overline{\boldsymbol{y}})) \wedge (\boldsymbol{y} \notin \{100, 010, 001\} \vee (\overline{\boldsymbol{x}_\psi} \wedge (\boldsymbol{y} = 001))) \wedge (\neg(\boldsymbol{x}_\psi) \vee \overline{(\boldsymbol{y} = 110)})$, thus $\overline{T_\psi(\boldsymbol{x}_\psi)} \wedge (\boldsymbol{y} = 001) \equiv \overline{\boldsymbol{x}_\psi} \wedge (\boldsymbol{y} = 001)$, and the second line becomes

$$\vee \left(\boldsymbol{y} \in \{100, 010, 001\} \wedge \neg\left(\overline{\boldsymbol{x}_{\psi_1}} \wedge \overline{\boldsymbol{x}_{\psi_2}} \wedge (\boldsymbol{y} = 001)\right)\right).$$

Since $T_\psi(\boldsymbol{x}_\psi) \wedge (\boldsymbol{y} = 110) \equiv (\boldsymbol{x}_\psi \wedge (\boldsymbol{y} = 110))$ for any ψ, the third line becomes

$$\vee (\boldsymbol{x}_{\psi_1} \wedge \boldsymbol{x}_{\psi_2} \wedge (\boldsymbol{y} = 110)).$$

Now $T_\wedge[x_1/T_{\psi_1}, x_2/T_{\psi_2}]$ equals

$$\begin{aligned} T_{\psi_1 \wedge \psi_2} = {} & (\psi_1(\boldsymbol{x}_{\psi_1}) \wedge \psi_2(\boldsymbol{x}_{\psi_2}) \wedge \boldsymbol{y}) \vee \left(\overline{\psi_1(\overline{\boldsymbol{x}_{\psi_1}}) \wedge \psi_2(\overline{\boldsymbol{x}_{\psi_2}})} \wedge \overline{\boldsymbol{y}}\right) \\ & \vee \left(\boldsymbol{y} \in \{100, 010, 001\} \wedge \neg\left(\overline{\boldsymbol{x}_{\psi_1}} \wedge \overline{\boldsymbol{x}_{\psi_2}} \wedge (\boldsymbol{y} = 001)\right)\right) \\ & \vee (\boldsymbol{x}_{\psi_1} \wedge \boldsymbol{x}_{\psi_2} \wedge (\boldsymbol{y} = 110)). \end{aligned}$$

2. This follows from 1. by Proposition 1.

3. By Lemma 2 there is an 1-reproducing $(n-3)$-ary function f with diameter of at least $2^{\lfloor \frac{n-3}{2} \rfloor}$. Let f be represented by a formula ϕ; then, T_ϕ represents an n-ary function of the same diameter in D_1. □

Lemma 8. *If* $[B] \supseteq \mathsf{S}_{02}^k$,

1. ST-BF-CONN(B) *and* BF-CONN(B) *are* PSPACE-*complete,*
2. ST-CIRC-CONN(B) *and* CIRC-CONN(B) *are* PSPACE-*complete,*
3. *for* $n \geq k+4$, *there is an* n*-ary function* $f \in [B]$ *with diameter of at least* $2^{\lfloor \frac{n-k-2}{2} \rfloor}$.

Proof. 1. This proof is analogous to the previous one. For a 1-reproducing formula ψ over the n variables $x_1, \ldots, x_n$, we construct the formula $T_\psi^\sim \in \mathsf{S}_{02}^k$ with the additional variables y and $(z_1, \ldots, z_{k+1}) = \boldsymbol{z}$,

$$T_\psi^\sim = (\psi \wedge y \wedge \overline{\boldsymbol{z}}) \vee \boldsymbol{z} \notin \{0\cdots0, 10\cdots0, 010\cdots0, \ldots, 0\cdots01\}.$$

$T_\psi^\sim(\boldsymbol{x}, y, \boldsymbol{z})$ is 0-separating of degree k since all vectors that are no solutions of $T_\psi^\sim$ end with a vector $\boldsymbol{b} \in \{0\cdots0, 10\cdots0, 010\cdots0, \ldots, 0\cdots01\} \subset \{0,1\}^{k+1}$ and thus any k of them have at least one common variable assigned 0. Also, $T_\psi^\sim$ is 0-reproducing and still 1-reproducing.

Further, for any two solutions $\boldsymbol{s}$ and $\boldsymbol{t}$ of $\psi(\boldsymbol{x})$, $\boldsymbol{s}' = \boldsymbol{s} \cdot 1 \cdot 0\cdots0$ and $\boldsymbol{t}' = \boldsymbol{t} \cdot 1 \cdot 0\cdots0$ are solutions of $T_\psi^\sim(\boldsymbol{x}, y, \boldsymbol{z})$ and are connected in $G(T_\psi^\sim)$ iff $\boldsymbol{s}$ and $\boldsymbol{t}$ are connected in $G(\psi)$.

But again, we have produced an additional connected component (consisting of all vectors not ending with $10\cdots0, 010\cdots0, \ldots, 0\cdots01$, or $0\cdots0$). To connect it to a component corresponding to one of ψ, we add $1\cdots1{\cdot}1{\cdot}10\cdots0$ as a solution,

$$\begin{aligned} T_\psi = {} & (\psi \wedge y \wedge \overline{\boldsymbol{z}}) \vee \boldsymbol{z} \notin \{0\cdots0, 10\cdots0, 010\cdots0, \ldots, 0\cdots01\} \\ & \vee(\boldsymbol{x} \wedge y \wedge (\boldsymbol{z} = 10\cdots0)). \end{aligned}$$

Now $G(T_\psi)$ is connected iff $G(\psi)$ is connected.

Again we show that the algorithm TR works in this case. Here,

$$\begin{aligned} T_\wedge[x_1/T_{\psi_1}, x_2/T_{\psi_2}] = {} & (T_{\psi_1}(\boldsymbol{x}_{\psi_1}) \wedge T_{\psi_2}(\boldsymbol{x}_{\psi_2}) \wedge y \wedge \overline{\boldsymbol{z}}) \\ & \vee \boldsymbol{z} \notin \{0\cdots0, 10\cdots0, 010\cdots0, \ldots, 0\cdots01\} \\ & \vee (T_{\psi_1}(\boldsymbol{x}_{\psi_1}) \wedge T_{\psi_2}(\boldsymbol{x}_{\psi_2}) \wedge y \wedge (\boldsymbol{z} = 10\cdots0)), \end{aligned}$$

which is equivalent to

$$\begin{aligned} T_{\psi_1 \wedge \psi_2} = {} & (\psi_1(\boldsymbol{x}_{\psi_1}) \wedge \psi_2(\boldsymbol{x}_{\psi_2}) \wedge y \wedge \overline{\boldsymbol{z}}) \\ & \vee \boldsymbol{z} \notin \{0\cdots0, 10\cdots0, 010\cdots0, \ldots, 0\cdots01\} \\ & \vee (\boldsymbol{x}_{\psi_1} \wedge \boldsymbol{x}_{\psi_2} \wedge y \wedge (\boldsymbol{z} = 10\cdots0)). \end{aligned}$$

2. This follows from 1. by Proposition 1.

3. By Lemma 2 there is an 1-reproducing $(n-k-2)$-ary function f with diameter of at least $2^{\lfloor \frac{n-k-2}{2} \rfloor}$. Let f be represented by a formula ϕ; then, T_ϕ represents an n-ary function of the same diameter in S_{02}^k. □

This completes the proof of Theorem 1.

5 The Connectivity of Quantified Formulas

Definition 3. *A* quantified B-formula ϕ *(in prenex normal form) is an expression of the form*

$$Q_1 y_1 \cdots Q_m y_m \varphi(y_1, \ldots, y_m, x_1, \ldots x_n),$$

where φ *is a* B*-formula, and* $Q_1, \ldots, Q_m \in \{\exists, \forall\}$ *are quantifiers.* $x_1, \ldots, x_n$ *are called the free variables of* ϕ*.*

For quantified B-formulas, we define the connectivity problems

- QBF-Conn(B): Given a quantified B-formula ϕ, is $G(\phi)$ connected?
- st-QBF-Conn(B): Given a quantified B-formula ϕ and two solutions $\boldsymbol{s}$ and $\boldsymbol{t}$, is there a path from $\boldsymbol{s}$ to $\boldsymbol{t}$ in $G(\phi)$?

Theorem 2. *Let* B *be a finite set of Boolean functions.*

1. *If* $B \subseteq \mathsf{M}$ *or* $B \subseteq \mathsf{L}$*, then*
   (a) st-QF-Conn(B) *and* QBF-Conn(B) *are in* P*,*
   (b) *the diameter of every quantified* B*-formula is linear in the number of free variables.*
2. *Otherwise,*
   (a) st-QBF-Conn(B) *and* QBF-Conn(B) *are* PSPACE*-complete,*
   (b) *there are quantified* B*-formulas with at most one quantifier such that their diameter is exponential in the number of free variables.*

Proof. See the extended version of this paper [16].

Remark 1. An analog to Theorem 2 also holds for quantified circuits as defined in [12, Section 7].

6 Conclusions

While the classification for CSPs required an essential enhancement of Schaefer's framework and the introduction of new classes of CNF($\mathcal{S}$)-formulas, for B-formulas and B-circuits the connectivity issues fit entirely into Post's framework, although the proofs were quite novel, and made substantial use of Gopalan et al.'s results for 3-CNF-formulas.

As Gopalan et al. stated, we also believe that "connectivity properties of Boolean satisfiability merit study in their own right", which is substantiated by the recent interest in reconfiguration problems. Moreover, we imagine our results could aid the advancement of circuit based SAT solvers.

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

References

1. Böhler, E., Creignou, N., Reith, S., Vollmer, H.: Playing with boolean blocks, part i: Posts lattice with applications to complexity theory. In: SIGACT News (2003)
2. Fu, Z., Malik, S.: Extracting logic circuit structure from conjunctive normal form descriptions. In: 20th International Conference on VLSI Design, Held Jointly with 6th International Conference on Embedded Systems, pp. 37–42. IEEE (2007)
3. Gopalan, P., Kolaitis, P.G., Maneva, E., Papadimitriou, C.H.: The connectivity of boolean satisfiability: Computational and structural dichotomies. SIAM J. Comput. 38(6), 2330–2355 (2009), http://dx.doi.org/10.1137/07070440X
4. Ito, T., Demaine, E.D., Harvey, N.J.A., Papadimitriou, C.H., Sideri, M., Uehara, R., Uno, Y.: On the complexity of reconfiguration problems. Theor. Comput. Sci. 412(12-14), 1054–1065 (2011), http://dx.doi.org/10.1016/j.tcs.2010.12.005
5. Kamiński, M., Medvedev, P., Milanič, M.: Shortest paths between shortest paths and independent sets. In: Iliopoulos, C.S., Smyth, W.F. (eds.) IWOCA 2010. LNCS, vol. 6460, pp. 56–67. Springer, Heidelberg (2011)
6. Lewis, H.R.: Satisfiability problems for propositional calculi. Mathematical Systems Theory 13(1), 45–53 (1979)
7. Makino, K., Tamaki, S., Yamamoto, M.: On the boolean connectivity problem for horn relations. In: Marques-Silva, J., Sakallah, K.A. (eds.) SAT 2007. LNCS, vol. 4501, pp. 187–200. Springer, Heidelberg (2007)
8. Maneva, E., Mossel, E., Wainwright, M.J.: A new look at survey propagation and its generalizations. Journal of the ACM (JACM) 54(4), 17 (2007)
9. Mézard, M., Mora, T., Zecchina, R.: Clustering of solutions in the random satisfiability problem. Physical Review Letters 94(19), 197205 (2005)
10. Michael, T.: On the applicability of post's lattice. Information Processing Letters 112(10), 386–391 (2012)
11. Post, E.L.: The Two-Valued Iterative Systems of Mathematical Logic(AM-5), vol. 5. Princeton University Press (1941)
12. Reith, S., Wagner, K.W.: The complexity of problems defined by Boolean circuits (2000)
13. Schaefer, T.J.: The complexity of satisfiability problems. In: STOC 1978, pp. 216–226 (1978)
14. Schnoor, H.: Algebraic techniques for satisfiability problems. Ph.D. thesis, Universität Hannover (2007)
15. Schwerdtfeger, K.W.: A computational trichotomy for connectivity of boolean satisfiability. ArXiv CoRR abs/1312.4524 (2013), extended version of a paper submitted to the JSAT Journal, http://arxiv.org/abs/1312.4524
16. Schwerdtfeger, K.W.: The connectivity of boolean satisfiability: Dichotomies for formulas and circuits. ArXiv CoRR abs/1312.6679 (2013), extended version of this paper, http://arxiv.org/abs/1312.6679
17. Vollmer, H.: Introduction to Circuit Complexity: A Uniform Approach. Springer-Verlag New York, Inc. (1999)
18. Wu, C.A., Lin, T.H., Lee, C.C., Huang, C.Y.R.: Qutesat: a robust circuit-based sat solver for complex circuit structure. In: Proceedings of the Conference on Design, Automation and Test in Europe, EDA Consortium, pp. 1313–1318 (2007)
19. Zverovich, I.E.: Characterizations of closed classes of boolean functions in terms of forbidden subfunctions and post classes. Discrete Appl. Math. 149(1-3), 200–218 (2005), http://dx.doi.org/10.1016/j.dam.2004.06.028