



HAL
open science

Computational Complexity of Minimal Trap Spaces in Boolean Networks

Kyungduk Moon, Kangbok Lee, Loïc Paulevé

► **To cite this version:**

Kyungduk Moon, Kangbok Lee, Loïc Paulevé. Computational Complexity of Minimal Trap Spaces in Boolean Networks. *SIAM Journal on Discrete Mathematics*, 2024, 38 (4), pp.2691-2708. 10.1137/23M1553248 . hal-04309014

HAL Id: hal-04309014

<https://hal.science/hal-04309014v1>

Submitted on 23 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

COMPUTATIONAL COMPLEXITY OF MINIMAL TRAP SPACES IN BOOLEAN NETWORKS*

KYUNGDUK MOON[†], KANGBOK LEE[†], AND LOIČ PAULEVÉ[‡]

Abstract. A Boolean network (BN) is a discrete dynamical system defined by a Boolean function that maps to the domain itself. A trap space of a BN is a generalization of a fixed point, which is defined as the sub-hypercubes closed by the function of the BN. A trap space is minimal if it does not contain any smaller trap space. Minimal trap spaces have applications for the analysis of attractors of BNs with various update modes. This paper establishes the computational complexity results of three decision problems related to minimal trap spaces: the decision of the trap space property of a sub-hypercube, the decision of its minimality, and the decision of the membership of a given configuration to a minimal trap space. Under several cases on Boolean function representations, we investigate the computational complexity of each problem. In the general case, we demonstrate that the trap space property is coNP-complete, and the minimality and the membership properties are Π_2^P -complete. The complexities drop by one level in the polynomial hierarchy whenever the local functions of the BN are eitherunate, or are represented using truth-tables, binary decision diagrams, or double DNFs (Petri net encoding): the trap space property can be decided in a polynomial time, whereas deciding the minimality and the membership are coNP-complete. When the BN is given as its functional graph, all these problems are in P.

Key words. Automata network, Trap space, Computational complexity, Boolean function representation, System dynamics, Attractors

MSC codes. 68Q17, 68R07, 94C11, 37M22, 37N25

1. Introduction. A Boolean network (BN) is a dynamical system defined by a function f of the Boolean domain with a fixed dimension n that maps to the domain itself, i.e., $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ with $\mathbb{B} = \{0, 1\}$. The function mapping to a component of the image of f is called a *local function*. We denote the local function mapping to the i -th component of the image as $f_i : \mathbb{B}^n \rightarrow \mathbb{B}$ for $i \in \{1, \dots, n\}$. Given a Boolean vector $\mathbf{x} \in \mathbb{B}^n$ referred to as a *configuration*, one can define a set of succeeding configurations by f following an *update mode* [3, 14, 25, 34], leading to a dynamical system. For instance, the synchronous (or parallel) update mode associates $f(\mathbf{x})$ as the unique succeeding configuration of \mathbf{x} by f , while the fully asynchronous mode considers any configuration resulted from the update of a single component as successors, leading to non-deterministic dynamics. BNs are studied in various disciplines such as discrete mathematics [2, 28] and dynamical system theory [5, 12, 24]. They also have wide applications to the modeling of complex systems such as biological systems [1, 15, 18, 31, 37], and social behaviors [13, 27], to name but a few.

The literature addresses a vast zoo of update modes, generating possibly different dynamics from the same BN f [26]. In this context, the dynamical properties of BNs which are independent of the update mode are of particular interest because they

*The authors certify that this is an original research article submitted to SIAM Journal on Discrete Mathematics on February 20, 2023.

Funding: KM and KL acknowledges support from the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (grant number: NRF-2022R1F1A107414011) LP acknowledges support from CampusFrance in the scope of PHC STAR 2024 project 50147NM and the French Agence Nationale pour la Recherche (ANR) in the scope of the project BNeDiction (grant number: ANR-20-CE45-0001).

[†]Department of Industrial and Management Engineering, POSTECH, South Korea (kaleb.moon@postech.ac.kr, kblee@postech.ac.kr).

[‡]Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800 F-33400 Talence, France (loic.pauleve@labri.fr)

show inherent dynamical properties of a given BN. The prime example is the study of fixed points of f , i.e., the configurations \mathbf{x} such that $f(\mathbf{x}) = \mathbf{x}$. Indeed, a fixed point of f can be assumed to be a stable state of dynamics that does not change after a transition with any update mode. Conversely, a stable state under the synchronous or the asynchronous update mode is a fixed point. Nevertheless, some specific update modes may exhibit additional stable states [26]. The notion of fixed points of f can be generalized to *trap spaces*. A trap space is a sub-hypercube (an n -dimensional sub-graph of the n -dimensional hypercube where some dimensions can be fixed to be singular values) closed by f such that for any vertex \mathbf{x} of the trap space, $f(\mathbf{x})$ is also one of its vertices. A *minimal trap space* is a trap space that contains no other trap spaces. Therefore, a fixed point of f is a specific case of minimal trap spaces where all dimensions are fixed.

It is important to remark that trap spaces are independent of the update mode [23]. Nevertheless, minimal trap spaces have also been studied with regard to *attractors*, which depend on the update mode and are important features of the long-term dynamical properties of BNs [15]. Given an update mode, an attractor is defined as an inclusion-wise minimal set of configurations which are closed by transitions. Equivalently, an attractor is a set of configurations satisfying the following two conditions. First, there exists a sequence of transitions between any pair of its configurations. Second, if there exists a sequence of transitions from one of its configuration \mathbf{x} to another configuration \mathbf{x}' , then they belong to the same attractor. If an attractor is composed by a single configuration, it is a fixed point; otherwise, it is called a *cyclic attractor*. It appears that any minimal trap space necessarily encloses at least one attractor of any update mode [16, 23, 26]. Moreover, minimal trap space that are not fixed points enclose necessarily at least one (a)synchronous cyclic attractors. Actually, for a large range of models of biological networks, minimal trap spaces approximate well their asynchronous attractors [17]. Beside the synchronous and the asynchronous update mode, minimal trap spaces are exactly the attractors of BNs under the most permissive update mode [23, 25], which guarantees to capture all transitions realized by any multi-valued refinement of the BN.

So far, the literature has essentially focused on algorithms and implementations for enumerating minimal trap spaces of BNs [16, 23, 35]. Nevertheless, whereas these algorithms indicate upper bounds for the computational complexity of decision problems related to minimal trap spaces, no lower bound has been characterized. In this paper, we provide computational complexity results of problems related to the minimal trap spaces. We focus on three fundamental decisions problems:

TRAPSPACE(f, \mathbf{h}) :

Given a BN f and a sub-hypercube \mathbf{h} , \mathbf{h} is a trap space of f .

MINTRAP(f, \mathbf{h}) :

Given a BN f and a sub-hypercube \mathbf{h} , \mathbf{h} is a minimal trap space of f .

IN-MINTRAP(f, \mathbf{x}) :

Given a BN f and a configuration \mathbf{x} , \mathbf{x} is a vertex of a minimal trap space of f .

We study the computational complexity of these problems depending on Boolean function representations and on the unate property of local functions, as summarized in Table 1.1. The label “-c” indicates the completeness of the problem. In the case of Boolean functions represented as *propositional formulas* (PF), upper bounds of these three decisions problems have been determined in [23]: TRAPSPACE is in coNP, whereas MINTRAP and IN-MINTRAP are in Π_2^P . Moreover, whenever the BN is *locally monotone*, i.e., each of its local functions is unate (its expression does not contain

89 a variable appearing both positively and negatively), they showed that TRAPSPACE
 90 is in P, whereas MINTRAP and IN-MINTRAP are in coNP. We complete the results of
 91 [23] by demonstrating the lower bound results for each corresponding case. We further
 92 consider three representations for local functions: *truth tables* (TT), *binary decision*
 93 *diagrams* (BDD), and *double disjunctive normal forms* (DDNFs). These representa-
 94 tions have practical relevance since binary decision diagrams are frequently employed
 95 by software [20] and double DNFs are employed by Petri nets [7, 21, 35]. For all
 96 three representations, the same computational complexities are demonstrated as the
 97 locally monotone case. Finally, we also consider a BN represented by its *functional*
 98 *graph* (FG), which matches with the state transition graph under the synchronous
 99 update mode: this graph associates each state in \mathbb{B}^n to its image by f . Therefore, we
 100 consider two classes of representations of BNs: either by the representation of their
 101 local functions, or by the representation of the global function f .

TABLE 1.1
The computational complexity of decision problems related to trap spaces

Boolean network		Problems		
Encoding	Unate property	TRAPSPACE	MINTRAP	IN-MINTRAP
<i>Representation with local functions</i>				
PF	General	coNP-c [†]	Π_2^P -c [†]	Π_2^P -c [†]
PF	Locally monotone	P [†]	coNP-c [†]	coNP-c [†]
TT	General	P	coNP-c	coNP-c
BDD	General	P	coNP-c	coNP-c
DDNFs	General	P	coNP-c	coNP-c
<i>Representation with a state transition graph</i>				
FG	General	P	P	P

[†]the upper bound results are presented in [23]

102 The rest of this paper is organized as follows. In Sec. 2, we introduce notations
 103 and terminologies that are used to define the problems and explain the results. In
 104 Sec. 3, we provide computational complexity results of these problems in different
 105 BN settings. Sec. 4 provides a concluding remark.

106 **2. Preliminaries.** We denote integers ranging from 1 to n by $[1, n] := \{1, \dots, n\}$.
 107 We use an interval subscript of a vector to denote the list of components lying on the
 108 interval's range. For example, $\mathbf{x}_{[1, n_1]}$ denotes the vector concatenating the first n_1
 109 components of vector \mathbf{x} .

110 **2.1. Representations and the unate property of local functions in Bool-
 111 ean networks.** A Boolean function on n variables is of the form $\phi : \mathbb{B}^n \rightarrow \mathbb{B}$. In this
 112 context, the following decision problems are relevant to our study. The SAT problem
 113 is to decide whether there exists a configuration $\mathbf{x} \in \mathbb{B}^n$ such that $\phi(\mathbf{x}) = 1$. We can
 114 generalize SAT by partitioning variables and alternately putting quantifiers ‘ \exists ’ and
 115 ‘ \forall ’ on them. Given the number of quantifiers $l \in \mathbb{Z}^+$, and the number of indices of
 116 partitions n_1, \dots, n_l such that $n_j < n_{j+1}$ for each $j \in [1, l - 1]$, we can define the
 117 following generalization to Σ_l SAT:

118
$$\Sigma_l$$
SAT: decide if $\exists \mathbf{x}_{[1, n_1]} \forall \mathbf{x}_{[n_1+1, n_2]} \exists \mathbf{x}_{[n_2+1, n_3]} \cdots \phi(\mathbf{x}) = 1$.

119 Analogously, we can define Π_l SAT as the complementary problem of Σ_l SAT. Notably,
 120 Σ_l SAT and Π_l SAT are together called *the true quantified Boolean formula* problem,

121 which is well known as a complete problem in the polynomial hierarchy; we refer
 122 to [4] and [32] for the details. Another relevant class is P^{NP} , the group of prob-
 123 lems that are polynomial time solvable using an oracle for NP. It is well-known that
 124 $P^{NP} \subseteq \Sigma_2^P \cap \Pi_2^P$. In this paper, we limit our focus to SAT, Π_1 SAT, and Π_2 SAT. The
 125 computational complexity of these problems depends on the representation of ϕ and
 126 its unate property listed as follows.

127 *As propositional formula.* Boolean function ϕ can be represented as a proposi-
 128 tional formula, which consists of Boolean variables $\mathbf{x}_1, \dots, \mathbf{x}_n$ and logical connectives
 129 \wedge (conjunction), \vee (disjunction), and \neg (negation). The size of a formula is its length,
 130 which is proportional to the total count of variables and connectives appearing in the
 131 input string.

132 *As disjunctive normal form (DNF).* Boolean function ϕ can be represented as a
 133 propositional formula consisting of a disjunction of conjunctive clauses. Negations are
 134 allowed only on variables and not on clauses. A DNF can be equivalently represented
 135 as a list of sets of literals, where a literal is either a variable or a negated variable.
 136 Any propositional formula can be represented in DNF with clauses consisting of at
 137 most three literals (3DNF), although its size may be exponentially large.

138 *Unate (monotone) case.* Boolean function ϕ is *unate* if there exists an ordering
 139 of components $\preceq \in \{\leq, \geq\}^n$ such that $\forall \mathbf{x}, \mathbf{y} \in \mathbb{B}^n, \left((\mathbf{x}_1 \preceq_1 \mathbf{y}_1) \wedge \dots \wedge (\mathbf{x}_n \preceq_n \mathbf{y}_n) \right) \Rightarrow$
 140 $\phi(\mathbf{x}) \leq \phi(\mathbf{y})$. In other words, for each component $j \in [1, n]$ and every configuration
 141 $\mathbf{x} \in \mathbb{B}^n, \phi(\mathbf{x}_{[1, j-1]} 0 \mathbf{x}_{[j+1, n]}) \preceq_j \phi(\mathbf{x}_{[1, j-1]} 1 \mathbf{x}_{[j+1, n]})$ holds.

142 *As truth table (TT).* Boolean function ϕ can be encoded as binary vector t with
 143 2^n rows, where for each row $m \in [1, 2^n], t_m$ is the value of $f_i(\text{bin}(m-1))$ with $\text{bin}(m)$
 144 being the binary representation of m .

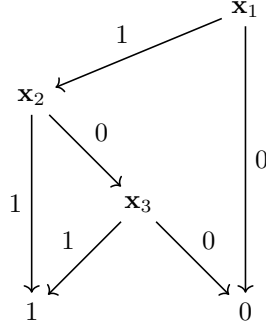
145 *As binary decision diagram (BDD).* A BDD has a directed acyclic graph structure
 146 with a unique root and at most two terminal nodes among 0 and 1 [10]. Each non-
 147 terminal node is associated to a component $i \in [1, n]$ and has two out-going edges,
 148 one labeled with 0 and the other with 1. Moreover, any path from the root to a
 149 terminal node crosses at most one node associated to each component. Then, each
 150 configuration \mathbf{x} corresponds to a single path from the root to a terminal node such
 151 that the edge emanating from a node associated with component i is labeled 1 if and
 152 only if $\mathbf{x}_i = 1$. This characterization captures common variants of BDDs, including
 153 reduced ordered BDDs [36].

154 *As double DNF (DDNF).* Boolean function ϕ can be represented with two DNFs
 155 ϕ^0 and ϕ^1 of n variables $\mathbf{x}_1, \dots, \mathbf{x}_n$ such that ϕ^0 is satisfied if and only if $\phi(\mathbf{x}) = 0$,
 156 and ϕ^1 is satisfied if and only if $\phi(\mathbf{x}) = 1$. This representation is typically employed
 157 in Petri nets [6, 7] and automata networks [21].

158 *Example 2.1.* We consider propositional formula $f = \mathbf{x}_1 \wedge \neg(\mathbf{x}_2 \wedge \neg\mathbf{x}_3)$ and show
 159 its different representation schemes and related explanations.

- 160 • $(\mathbf{x}_1 \wedge \neg\mathbf{x}_2) \vee (\mathbf{x}_1 \wedge \mathbf{x}_3)$ is an equivalent DNF representation of f .
- 161 • f is unate with \preceq being a vector $\langle \leq, \geq, \leq \rangle \in \{\leq, \geq\}^3$. On the other hand,
 162 another Boolean formula $(\neg\mathbf{x}_1 \wedge \mathbf{x}_2) \vee (\mathbf{x}_1 \wedge \neg\mathbf{x}_2)$ is not unate.
- 163 • The truth table representation of f is $t = 00001101$, assuming that $\text{bin}(1) =$
 164 001 .
- 165 • One of the double DNF representations of f is (ϕ^0, ϕ^1) with $\phi^0 = (\neg\mathbf{x}_1) \vee$
 166 $(\mathbf{x}_2 \wedge \neg\mathbf{x}_3)$ and $\phi^1 = (\mathbf{x}_1 \wedge \neg\mathbf{x}_2) \vee (\mathbf{x}_1 \wedge \mathbf{x}_3)$;
- 167 • An equivalent BDD representation of f is the graph in Figure 2.1:

168 We use some known computational complexity results in the literature as follows.


 FIG. 2.1. A binary decision diagram representation of the Boolean network f in Example 2.1

169 If ϕ is a general propositional formula, SAT is NP-c, Π_1 SAT is coNP-c, and Π_2 SAT is
 170 Π_2 -c [4]. If ϕ is a DNF or a 3DNF, SAT is P, Π_1 SAT is coNP-c [8]. If ϕ is represented
 171 as a locally monotone propositional formula, TT, BDD, or DDNFs, both SAT and
 172 Π_1 SAT are in P [8, 36].

173 *Boolean networks (BNs).* Recall that a BN of dimension n is defined by a function
 174 $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ with its local (Boolean) function of the i -th component $f_i : \mathbb{B}^n \rightarrow \mathbb{B}$
 175 for $i \in [1, n]$. For a locally monotone BN, we assume that the orderings of compo-
 176 nents leading to their unate property are given. The local Boolean functions of the
 177 BN can be encoded with any of the aforementioned representations. In the case of
 178 truth tables, the dimension of the truth table a local function follows the number of
 179 components which it depends on. A function f_i depends on component j if there
 180 exists a configuration $\mathbf{y} \in \mathbb{B}^n$ such that $f_i(\mathbf{y}_{[1, j-1]} \mathbf{0}_{[j+1, n]}) \neq f_i(\mathbf{y}_{[1, j-1]} \mathbf{1}_{[j+1, n]})$.
 181 In practice, we can significantly reduce the dimension of f_i from n in the following
 182 way. If k is the number of components that f_i depends on, we define its corresponding
 183 integer vector $p \in [1, n]^k$ to list up the indices of such components in the BN. Then,
 184 a truth table t with 2^k rows can be constructed to satisfy $f_i(\mathbf{x}) = t_{\mathbf{x}_{p_1} \dots \mathbf{x}_{p_k}}$ for any
 185 configuration $\mathbf{x} \in \mathbb{B}^n$ of the BN. Finally, a BN can be represented by its *functional*
 186 *graph*, the digraph of the image by f . It is also known as the synchronous state tran-
 187 sition graph. The vertices of such a graph are all the configurations \mathbb{B}^n , and there is
 188 an edge from \mathbf{x} to \mathbf{y} if and only if $\mathbf{y} = f(\mathbf{x})$.

189 *Example 2.2.* The BN $f : \mathbb{B}^3 \rightarrow \mathbb{B}^3$ with

$$\begin{aligned} 190 \quad f_1(\mathbf{x}) &= (\neg \mathbf{x}_1 \vee \neg \mathbf{x}_2) \wedge \mathbf{x}_3 \\ 191 \quad f_2(\mathbf{x}) &= \mathbf{x}_1 \wedge \mathbf{x}_3 \\ 192 \quad f_3(\mathbf{x}) &= \mathbf{x}_1 \vee \mathbf{x}_2 \vee \mathbf{x}_3 \end{aligned}$$

193 is locally monotone since all its local functions are unate. The functional graph of f
 194 is illustrated in Figure 2.2.

195 **2.2. Sub-hypercubes and minimal trap spaces of BNs.** A sub-hypercube
 196 is an n -dimensional sub-graph of the n -hypercube such that some dimensions can
 197 be fixed to be singular values. It can be represented as a vector $\mathbf{h} \in \{0, 1, *\}^n$,
 198 which specifies for each dimension $i \in [1, n]$ whether it is at a fixed value (0 or 1),
 199 or free (*) in the sub-hypercube. The vertices of a sub-hypercube \mathbf{h} are denoted by
 200 $v(\mathbf{h}) := \{\mathbf{x} \in \mathbb{B}^n : \forall i \in [1, n], (\mathbf{h}_i \neq *) \Rightarrow (\mathbf{x}_i = \mathbf{h}_i)\}$. A sub-hypercube \mathbf{h} is smaller

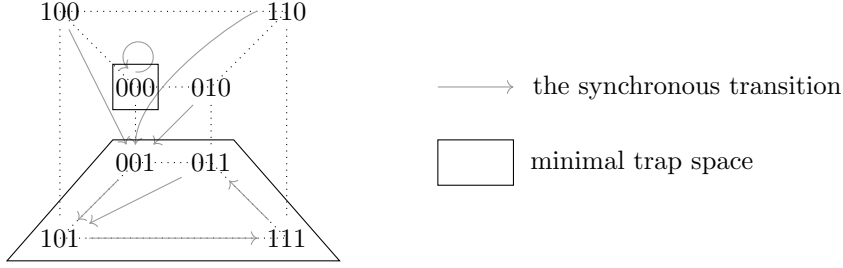


FIG. 2.2. The functional graph and minimal trap spaces of the Boolean network f in Example 2.2

201 than a sub-hypercube space \mathbf{h}' whenever $v(\mathbf{h}) \subseteq v(\mathbf{h}')$. We also write this condition
 202 as $\mathbf{h} \subseteq \mathbf{h}'$.

203 A trap space of a BN f is a sub-hypercube $\mathbf{h} \in \{0, 1, *\}^n$ which is *closed* by f ,
 204 i.e., for each vertex $\mathbf{x} \in v(\mathbf{h})$, $f(\mathbf{x}) \in v(\mathbf{h})$ implying the its image by f is also a vertex
 205 of \mathbf{h} . Remark that $*^n$ is always a trap space. A trap space \mathbf{h} is *minimal* if there is
 206 no different trap space $\mathbf{h}' \neq \mathbf{h}$ within itself; i.e., there exists no trap space \mathbf{h}' such
 207 that $v(\mathbf{h}') \subsetneq v(\mathbf{h})$. We use $\mathbf{T}(\mathbf{h})$ to denote the minimal trap space that contains all
 208 configurations in $v(\mathbf{h})$. In other words, $\mathbf{T}(\mathbf{h})$ must satisfy three properties:

- 209 • $\mathbf{T}(\mathbf{h})$ is a trap space,
- 210 • $\mathbf{h} \subseteq \mathbf{T}(\mathbf{h})$,
- 211 • There exists no trap space \mathbf{h}' such that $\mathbf{h} \subseteq \mathbf{h}' \subsetneq \mathbf{T}(\mathbf{h})$.

212 Remark that if \mathbf{h} is a minimal trap space, then, for any configuration $\mathbf{x} \in v(\mathbf{h})$,
 213 $\mathbf{T}(\mathbf{x}) = \mathbf{h}$.

214 *Example 2.3.* The BN f of *Example 2.2* has a fixed point $\{000\}$ and a cyclic
 215 attractor $\{011, 101, 111\}$. See Figure 2.2 for its functional graph representation and
 216 minimal trap spaces. It has two minimal trap spaces: 000 and $* * 1$. Moreover,
 217 $\mathbf{T}(010) = \mathbf{T}(01*) = \mathbf{T}(0 * 0) = * * *$.

218 **2.3. Upper bounds results to the computational complexity.** We present
 219 all the upper bound results when local functions are given. All the polynomial time
 220 solvable cases in Table 1.1 are also discussed here, except the ones with a functional
 221 graph. We later present polynomial time algorithms for the remaining cases in Sec.
 222 3.4. The basic ideas and previous upper bound results are adopted from [23], yet
 223 with some extensions to the representations we are considering. All new results are
 224 summarized in Theorem 2.4.

225 Consider NOT-TRAPSPACE(f, \mathbf{h}), the problem of deciding if the given hyper-
 226 cube \mathbf{h} is *not* closed by f : it is equivalent to deciding if there exists component
 227 $i \in [1, n]$ with $\mathbf{h}_i \neq *$ and $\mathbf{z} \in v(\mathbf{h})$ such that $f_i(\mathbf{z}) \neq \mathbf{h}_i$, which boils down to SAT.
 228 Thus, the complementary problem TRAPSPACE(f, \mathbf{h}) is in coNP for the general case
 229 and in P for the locally monotone case. For the same reason, TRAPSPACE(f, \mathbf{h}) is
 230 in P when the local functions are given as truth tables, BDDs, or double DNFs.

231 Now, consider NOT-MINTRAP(f, \mathbf{h}), the problem of deciding if the hypercube
 232 \mathbf{h} is either *not* closed by f or is closed but *not* minimal. It can be decided by first
 233 checking if \mathbf{h} is a trap space and then checking the existence of another trap space \mathbf{h}'
 234 which is strictly included in \mathbf{h} . This problem is at most $\text{NP}^{\text{TRAPSPACE}}$ because only
 235 the inclusion $\mathbf{h}' \subseteq \mathbf{h}$ needs to be decided with an oracle for TRAPSPACE, and it can
 236 be done in a polynomial time. Thus, the complementary problem MINTRAP(f, \mathbf{h})

237 is in $\text{coNP}^{\text{TRAPSPACE}}$, which is at most $\text{coNP}^{\text{coNP}} = \Pi_2^{\text{P}}$ in the general case. For
 238 the locally monotone case, $\text{MINTRAP}(f, \mathbf{h})$ is in coNP because TRAPSPACE can be
 239 solved in a polynomial time. For the same reason, $\text{MINTRAP}(f, \mathbf{h})$ is in coNP when
 240 the local functions are represented as truth tables, BDDs, or double DNFs.

241 Finally, consider $\text{IN-MINTRAP}(f, \mathbf{x})$ the problem of deciding whether the con-
 242 figuration \mathbf{x} is a vertex of a minimal trap space of f . It boils down to decide
 243 $\text{MINTRAP}(f, \mathbf{T}(\mathbf{x}))$. The computation of $\mathbf{T}(\mathbf{x})$ can be performed using Algorithm 2.1. ■

Algorithm 2.1 The minimal trap space containing a configuration

Input: Local functions f , the initial configuration \mathbf{x}
Output: The minimal trap space containing \mathbf{x} ($:= \mathbf{T}(\mathbf{x})$)

```

1:  $\mathbf{h} := \mathbf{x}$ 
2: for  $k \in [1, n]$  do
3:   for  $i \in [1, n]$  with  $\mathbf{h}_i \neq *$  do
4:     if  $\exists \mathbf{y} \in v(\mathbf{h})$  s.t.  $f_i(\mathbf{y}) = 1 - \mathbf{y}_i$  then
5:        $\mathbf{h}_i := *$ 
6: return  $\mathbf{h}$ 
    
```

244
 245 The procedure to check the existence in line 4 is equivalent to SAT. Thus, overall,
 246 this algorithm is in P^{NP} in the general case, and in P for the locally monotone case.
 247 Analogously, $\mathbf{T}(\mathbf{x})$ can be computed in a polynomial time when the local functions are
 248 represented as truth tables, BDDs, or double DNFs. For all cases, the computational
 249 complexity for computing $\mathbf{T}(\mathbf{x})$ does not exceed that of MINTRAP . Therefore, the
 250 computational complexity of IN-MINTRAP is up to the complexity of MINTRAP for
 251 each.

252 **THEOREM 2.4.** *Given hypercube \mathbf{h} and BN f with its local functions represented*
 253 *as truth tables, BDDs, or double DNFs, $\text{TRAPSPACE}(f, \mathbf{h})$ can be solved in a poly-*
 254 *nomial time.*

255 **3. Results.** In this section, we demonstrate computational complexity results
 256 for the TRAPSPACE , MINTRAP and IN-MINTRAP problems in BNs with different
 257 representations and the unate property. In Sec. 3.1, we present the exact computa-
 258 tional complexity for BNs with local functions given as propositional formulas, which
 259 is the most general case under our consideration. Results for the special case of locally
 260 monotone BNs are presented in Sec. 3.2. Those results are used in Sec. 3.3 to derive
 261 the computational complexity in the case of BNs with local functions represented
 262 with truth tables, binary decision diagrams, and double DNFs. The computational
 263 complexity for the BNs given as a functional graph is presented in Sec. 3.4.

264 **3.1. Local functions given as propositional formulas.** Theorem 3.1 demon-
 265 strates that TRAPSPACE is coNP -hard when local functions are represented as gen-
 266 eral propositional formulas, which is the lower bound to the computational complexity.
 267 Combined with the previous upper bound results of [23], the completeness is shown.
 268 The reduction scheme is illustrated in Figure 3.1 as an interaction graph of the vari-
 269 ables in the reduced problem. A green arc with an arrow represents that the source
 270 variable positively affects the local function of the target variable (i.e., marginally
 271 increasing the source variable never decreases the value of the local function). A red
 272 arc with a bar represents that the source variable negatively affect the local function
 273 of the target variable. A blue arc with a circle represents that the interaction may

274 be either positive or negative according to a Boolean function involved in the local
 275 function. Otherwise, variables are not involved in the local function. Variables are
 276 grouped to either match with the variables of the original SAT problem or to indicate
 277 they are auxiliary variables.

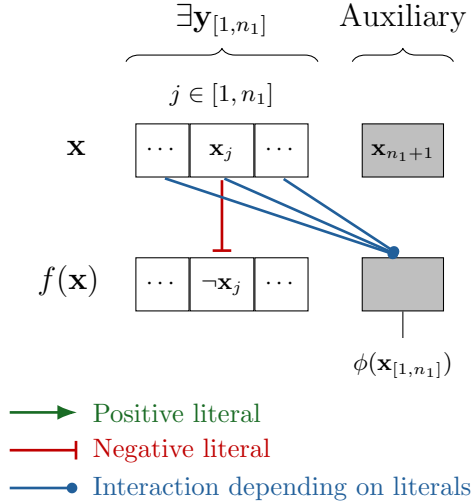


FIG. 3.1. Reduction from Π_1 SAT to TRAPSPACE for propositional formulas (*Theorem 3.1*).

278 **THEOREM 3.1.** *Given hypercube \mathbf{h} and BN f with its local functions represented*
 279 *as propositional formulas, $\text{TRAPSPACE}(f, \mathbf{h})$ is coNP-hard.*

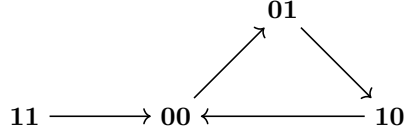
280 *Proof.* Consider a Boolean function $\phi : \mathbb{B}^{n_1} \rightarrow \mathbb{B}$ for $n_1 \in \mathbb{Z}^+$ and the associated
 281 Π_1 SAT problem of deciding $\forall \mathbf{y} \phi(\mathbf{y}) = 1$, which is coNP-complete. We construct BN
 282 $f : \mathbb{B}^{n_1+1} \rightarrow \mathbb{B}^{n_1+1}$ as

$$283 \quad \forall i \in [1, n_1], \quad f_i(\mathbf{x}) = \neg \mathbf{x}_i$$

$$284 \quad f_{n_1+1}(\mathbf{x}) = \phi(\mathbf{x}_{[1, n_1]})$$

285 and hypercube $\mathbf{h} = *^{n_1} \mathbf{1}$. We prove this theorem by showing that this Π_1 SAT problem
 286 is true if and only if $\text{TRAPSPACE}(f, \mathbf{h})$ is true. If $\text{TRAPSPACE}(f, \mathbf{h})$ is true, then
 287 $\forall \mathbf{z} \in v(\mathbf{h}), \phi(\mathbf{z}_{[1, n_1]}) = 1$. Since $\mathbf{z}_{[1, n_1]}$ can have an arbitrary configuration in \mathbb{B}^{n_1} ,
 288 $\forall \mathbf{y} \phi(\mathbf{y})$ must be true. On the other hand, if $\text{TRAPSPACE}(f, \mathbf{h})$ is false, we can
 289 find configuration $\mathbf{z} \in \mathbb{B}^{n_1}$ that satisfies $f_{n_1+1}(\mathbf{z}\mathbf{1}) = \phi(\mathbf{z}) = 0$. This can be used
 290 as a certificate that $\forall \mathbf{y} \phi(\mathbf{y})$ is not true, and it can be verified in a polynomial time.
 291 Hence, the theorem holds. \square

292 Theorem 3.6 and Theorem 3.7 demonstrate that MINTRAP and IN-MINTRAP
 293 are Π_2^P -hard, respectively. Combined with the previous upper bound results in [23],
 294 their completeness is shown. Our proofs show that MINTRAP and IN-MINTRAP
 295 can be used to solve Π_2^P SAT based on several tricks. First, a component \mathbf{x}_i with
 296 its local function $f_i = \neg \mathbf{x}_i$ always becomes free in a minimal trap space; see Remark
 297 3.2. We use this trick to encode Boolean variables quantified with \forall to the BN we
 298 construct. Second, given a Boolean formula ϕ to be proven its satisfiability, we employ
 299 two auxiliary components that have a full control to override other local functions as
 300 either 0 or 1 whenever ϕ is true. We use this trick to construct a BN that has


 FIG. 3.2. The transition graph of g from $\mathbf{z}_{[n+1, n+2]}$ to $g(\mathbf{z})_{[n+1, n+2]}$

301 the full dimensional hypercube as its unique minimal trap space if and only if ϕ is
 302 satisfied. Note that those auxiliary components will be always presented as the last
 303 two components of the BN we construct; see Remarks 3.3–3.4 for details. Figure 3.3
 304 illustrates the reduction scheme.

305 *Remark 3.2.* Let $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ with $n \in \mathbb{Z}^+$ be a BN. Given $\mathcal{I} \subseteq [1, n]$, suppose
 306 $f_i(\mathbf{x}) = \neg \mathbf{x}_i$ for $\forall i \in [1, n]$. Then, any hypercube $\mathbf{h} \in \{0, 1, *\}^n$ must satisfy $\mathbf{T}(\mathbf{h})_i = *$
 307 for all $i \in \mathcal{I}$.

308 *Proof.* For all $i \in \mathcal{I}$, component \mathbf{x}_i can be updated to $\neg \mathbf{x}_i$ and realized as both 0
 309 and 1. Therefore, $\mathbf{T}(\mathbf{h})_i = *$ for all $i \in \mathcal{I}$ to ensure that $\mathbf{T}(\mathbf{h})$ is closed by f . \square

310 *Remark 3.3.* For a given Boolean function $\phi : \mathbb{B}^n \rightarrow \mathbb{B}$ with $n \in \mathbb{Z}^+$, let $f :$
 311 $\mathbb{B}^{n+2} \rightarrow \mathbb{B}^{n+2}$ be a BN satisfying $\begin{cases} f_{n+1}(\mathbf{x}) = \phi(\mathbf{x}_{[1, n]}) \wedge \neg \mathbf{x}_{n+2} \\ f_{n+2}(\mathbf{x}) = \mathbf{x}_{n+1} \wedge \neg \mathbf{x}_{n+2} \end{cases}$. Suppose hy-
 312 percube $\mathbf{h} \in \{0, 1, *\}^{n+2}$ contains $\mathbf{z} \in v(\mathbf{h})$ that satisfies $\phi(\mathbf{z}_{[1, n]}) = 1$. Then,
 313 $\mathbf{T}(\mathbf{h})_{n+1} = \mathbf{T}(\mathbf{h})_{n+2} = *$.

314

315 *Proof.* Let $g : \mathbb{B}^{n+2} \rightarrow \mathbb{B}^{n+2}$ that maps $\mathbf{x} \mapsto \mathbf{x}_{[1, n]} f(\mathbf{x})_{[n+1, n+2]}$. Since $\mathbf{z}, f(\mathbf{z}) \in$
 316 $\mathbf{T}(\mathbf{h})$ by the definition of a trap space, $g(\mathbf{z}) = \mathbf{z}_{[1, n]} f(\mathbf{z})_{[n+1, n+2]} \in \mathbf{T}(\mathbf{h})$. Therefore,
 317 $\mathbf{T}(\mathbf{h})_{n+1} = \mathbf{T}(\mathbf{h})_{n+2} = *$ if there exists $k_1, k_2 \in \mathbb{Z}$ such that $g^{k_1}(\mathbf{z})_{n+1} = (1 -$
 318 $\mathbf{z}_{n+1})$ and $g^{k_2}(\mathbf{z})_{n+2} = (1 - \mathbf{z}_{n+2})$. Figure 3.2 shows projected transitions by g from
 319 $\mathbf{z}_{[n+1, n+2]}$ to $g(\mathbf{z})_{[n+1, n+2]}$ constructed using $\phi(\mathbf{z}_{[1, n]}) = 1$. Starting at any vector,
 320 there exists a transition path through which the $(n + 1)$ -th and $(n + 2)$ -th values are
 321 converted from the initial ones. Hence, the claim holds. \square

322 *Remark 3.4.* Let $f : \mathbb{B}^{n+2} \rightarrow \mathbb{B}^{n+2}$ with $n \in \mathbb{Z}^+$ be a BN. For some $i \in [1, n]$
 323 and a given Boolean function $\phi_i : \mathbb{B}^n \rightarrow \mathbb{B}$, suppose the local function $f_i(\mathbf{x})$ is in the
 324 form of $(\phi_i(\mathbf{x}_{[1, n]}) \wedge \neg \mathbf{x}_{n+1}) \vee \mathbf{x}_{n+2}$. If hypercube $\mathbf{h} \in \{0, 1, *\}^{n+2}$ satisfies $\mathbf{T}(\mathbf{h})_{n+1} =$
 325 $\mathbf{T}(\mathbf{h})_{n+2} = *$, then $\mathbf{T}(\mathbf{h})_i = *$.

326 *Proof.* Since $\mathbf{T}(\mathbf{h})_{n+1} = \mathbf{T}(\mathbf{h})_{n+2} = *$, there exists a configuration $\mathbf{z} \in v(\mathbf{T}(\mathbf{h}))$
 327 such that $(\mathbf{z}_{n+1}, \mathbf{z}_{n+2}) = (1, 0)$, which can be evaluated as $f_i(\mathbf{z}) = 0$. In addition, there
 328 exists another configuration $\mathbf{z}' \in v(\mathbf{T}(\mathbf{h}))$ such that $\mathbf{z}'_{n+2} = 1$, which can be evaluated
 329 as $f_i(\mathbf{z}') = 1$. Hence, the image of f_i can be both 0 and 1, implying $\mathbf{T}(\mathbf{h})_i = *$ to
 330 ensure that $\mathbf{T}(\mathbf{h})$ is closed by f . \square

331 **LEMMA 3.5.** Consider $n_1, n_2 \in \mathbb{Z}^+$ with $n_1 \leq n_2$ and a Boolean function $\phi :$
 332 $\mathbb{B}^{n_2} \rightarrow \mathbb{B}$ given as a propositional formula. Boolean formula $\forall \mathbf{y}_{[1, n_1]} \exists \mathbf{y}_{[n_1+1, n_2]} \phi(\mathbf{y})$
 333 is true if and only if BN $f : \mathbb{B}^{n_2+2} \rightarrow \mathbb{B}^{n_2+2}$ with the local functions defined by

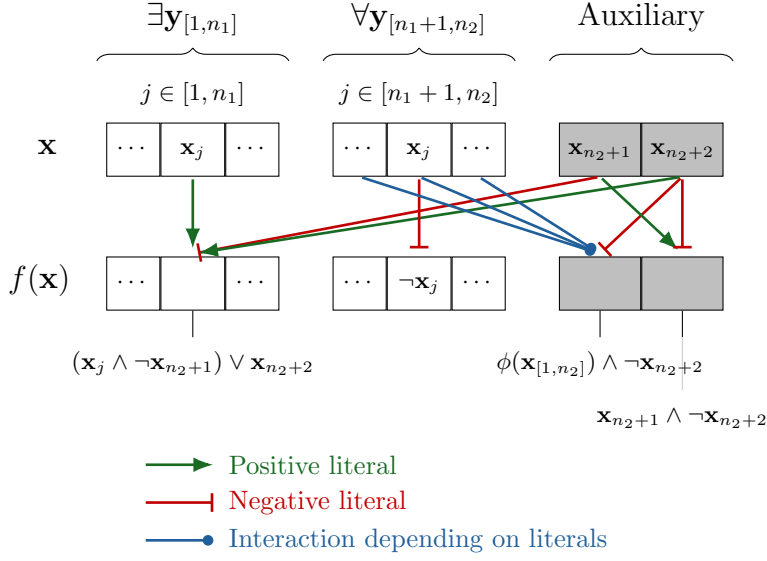


FIG. 3.3. Reduction from $\Pi_2\text{SAT}$ to MINTRAP and IN-MINTRAP for propositional formulas (Lemma 3.5).

334 (3.1)–(3.4) has the unique trap space $*^{n_2+2}$.

335 (3.1) $\forall j \in [1, n_1], \quad f_j(\mathbf{x}) = (\mathbf{x}_j \wedge \neg \mathbf{x}_{n_2+1}) \vee \mathbf{x}_{n_2+2}$

336 (3.2) $\forall j \in [n_1 + 1, n_2], \quad f_j(\mathbf{x}) = \neg \mathbf{x}_j$

337 (3.3) $f_{n_2+1}(\mathbf{x}) = \phi(\mathbf{x}_{[1, n_2]}) \wedge \neg \mathbf{x}_{n_2+2}$

338 (3.4) $f_{n_2+2}(\mathbf{x}) = \mathbf{x}_{n_2+1} \wedge \neg \mathbf{x}_{n_2+2}$

339 *Proof.* If $\forall \mathbf{y}_{[1, n_1]} \exists \mathbf{y}_{[n_1+1, n_2]} \phi(\mathbf{y})$ is true, any hypercube $\mathbf{h} \in \{0, 1, *\}^{n_2+2}$ satisfies

340 $\mathbf{T}(\mathbf{h}) \supseteq \mathbf{T}(\mathbf{h}_{[1, n_1]} *^{n_2-n_1} \mathbf{h}_{[n_2+1, n_2+2]}) \quad \because \text{Remark 3.2}$

341 $\supseteq \mathbf{T}(\mathbf{h}_{[1, n_1]} *^{n_2-n_1+2}) \quad \because \text{Remark 3.3}$

342 $\supseteq \mathbf{T}(*^{n_2+2}) \quad \because \text{Remark 3.4}$

343 $= *^{n_2+2}$

344 Therefore, $*^{n_2+2}$ is the unique (and thus minimal) trap space. For the remaining case
 345 where $\exists \mathbf{y}_{[1, n_1]} \forall \mathbf{y}_{[n_1+1, n_2]} \neg \phi(\mathbf{y})$, $*^{n_2+2}$ is not a minimal trap space because a smaller
 346 trap space $\mathbf{h}' = \mathbf{y}_{[1, n_1]} *^{n_2-n_1} \mathbf{0}^2$ exists. This completes the proof. \square

347 **THEOREM 3.6.** Given hypercube \mathbf{h} and BN f with its local functions represented
 348 as propositional formulas, $\text{MINTRAP}(f, \mathbf{h})$ is Π_2^P -hard.

349 *Proof.* Given $n_1, n_2 \in \mathbb{Z}^+$ with $n_1 \leq n_2$ and a Boolean function $\phi : \mathbb{B}^{n_2} \rightarrow \mathbb{B}$,
 350 consider the associated $\Pi_2\text{SAT}$ problem that deciding whether $\forall \mathbf{y}_{[1, n_1]} \exists \mathbf{y}_{[n_1+1, n_2]} \phi(\mathbf{y})$
 351 is true, which is Π_2^P -complete. By Lemma 3.5, this $\Pi_2\text{SAT}$ problem is true if and only
 352 if $\text{MINTRAP}(f, *^{n_2+2})$ is true for f defined by (3.1)–(3.4). Hence, the theorem holds. \square

353 **THEOREM 3.7.** Given configuration \mathbf{x} and BN f with its local functions repre-
 354 sented as propositional formulas, $\text{IN-MINTRAP}(f, \mathbf{x})$ is Π_2^P -hard.

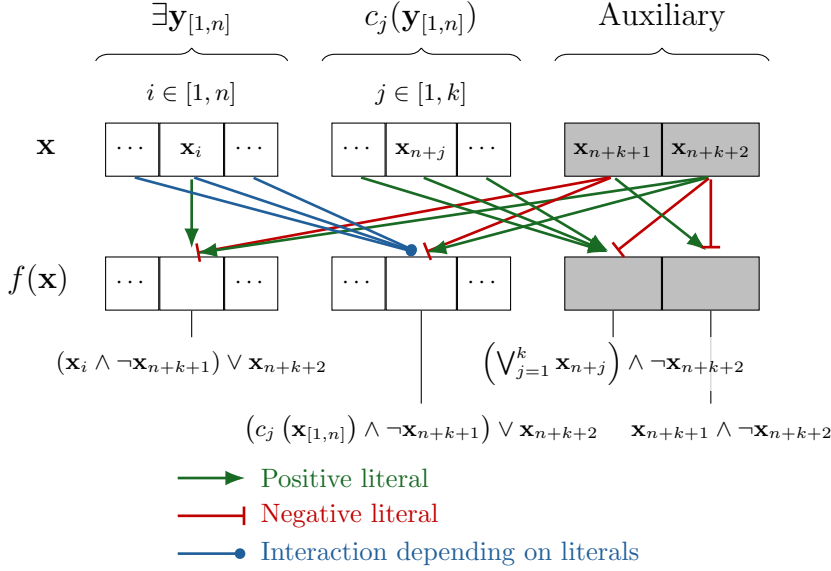


FIG. 3.4. Reduction from Π_1 SAT to MINTRAP and IN-MINTRAP for locally-monotone propositional formulas (Lemma 3.8).

355 *Proof.* Given $n_1, n_2 \in \mathbb{Z}^+$ with $n_1 \leq n_2$ and a Boolean function $\phi : \mathbb{B}^{n_2} \rightarrow \mathbb{B}$,
 356 consider the associated Π_2 SAT problem that deciding whether $\forall \mathbf{y}_{[1, n_1]} \exists \mathbf{y}_{[n_1+1, n_2]} \phi(\mathbf{y})$
 357 is true, which is Π_2^P -complete. We prove the theorem by showing that the Π_2 SAT is
 358 true if and only if IN-MINTRAP($f, \mathbf{1}^{n_2+2}$) is true for f defined by (3.1)–(3.4).

359 If $\forall \mathbf{y}_{[1, n_1]} \exists \mathbf{y}_{[n_1+1, n_2]} \phi(\mathbf{y})$, $*^{n_2+2}$ is the unique minimal trap space by Lemma
 360 3.5 and thus $\mathbf{1}^{n_2+2}$ belongs to a minimal trap space. For the remaining case where
 361 $\exists \mathbf{y}_{[1, n_1]} \forall \mathbf{y}_{[n_1+1, n_2]} \neg \phi(\mathbf{y})$, we have

$$\begin{aligned}
 362 \quad \mathbf{T}(\mathbf{1}^{n_2+2}) &\supseteq \mathbf{T}(\mathbf{1}^{n_1} *^{n_2-n_1} \mathbf{1}^2) && \because \text{Remark 3.2} \\
 363 &\supseteq \mathbf{T}(\mathbf{1}^{n_1} *^{n_2-n_1+2}) && \because x_{n_2+2} = 1 \\
 364 &\supseteq \mathbf{T}(*^{n_2+2}) && \because \text{Remark 3.4} \\
 365 &= *^{n_2+2}.
 \end{aligned}$$

366 However, $*^{n_2+2}$ is not a minimal trap space because there is a smaller trap space
 367 $\mathbf{h}' = \mathbf{y}_{[1, n_1]} *^{n_2-n_1} \mathbf{0}^2$. This completes the proof. \square

368 **3.2. Locally-monotone BNs with local functions given as propositional**
 369 **formulas.** We show a polynomial-time encoding of any DNF as a BN such that
 370 Π_1 SAT problem reduces to MINTRAP and IN-MINTRAP problems as illustrated
 371 in Figure 3.4. The proofs are given in Theorem 3.9 and 3.10, respectively. Let us
 372 consider any Boolean function $\phi : \mathbb{B}^n \rightarrow \mathbb{B}$ represented in DNF as a list of k conjunctive
 373 clauses. For $j \in [1, k]$, we use $c_j(\mathbf{y})$ to denote the j -th clause of ϕ evaluated with
 374 $\mathbf{y} \in \mathbb{B}^n$. Whenever $k = 0$, ϕ is considered to be false. Whenever a clause is empty
 375 it is equivalent to be true. We can assume that each clause $c_j(\mathbf{y})$ does not contain a
 376 contradiction caused by the same component (e.g., $y_i \wedge \neg y_i$). Therefore, all clauses
 377 are unate.

378 **LEMMA 3.8.** *Let us consider $n, k \in \mathbb{Z}^+$ and a Boolean function $\phi : \mathbb{B}^n \rightarrow \mathbb{B}$ given*

379 as a DNF with k conjunctive clauses; i.e., $\phi(\mathbf{y}) := \bigvee_{j=1}^k c_j(\mathbf{y})$. Boolean formula
 380 $\forall \mathbf{y} \phi(\mathbf{y})$ is true if and only if BN $f : \mathbb{B}^{n+k+2} \rightarrow \mathbb{B}^{n+k+2}$ with the local functions
 381 defined by (3.5)–(3.8) has the unique minimal trap space $*^{n+k+2}$.

$$382 \quad (3.5) \quad \forall i \in [1, n], \quad f_i(\mathbf{x}) = (\mathbf{x}_i \wedge \neg \mathbf{x}_{n+k+1}) \vee \mathbf{x}_{n+k+2}$$

$$383 \quad (3.6) \quad \forall j \in [1, k], \quad f_{n+j}(\mathbf{x}) = (c_j(\mathbf{x}_{[1,n]}) \wedge \neg \mathbf{x}_{n+k+1}) \vee \mathbf{x}_{n+k+2}$$

$$384 \quad (3.7) \quad f_{n+k+1}(\mathbf{x}) = \left(\bigvee_{j=1}^k \mathbf{x}_{n+j} \right) \wedge \neg \mathbf{x}_{n+k+2}$$

$$385 \quad (3.8) \quad f_{n+k+2}(\mathbf{x}) = \mathbf{x}_{n+k+1} \wedge \neg \mathbf{x}_{n+k+2}$$

386 *Proof.* If $\forall \mathbf{y} \phi(\mathbf{y})$ is true, any hypercube $\mathbf{h} \in \{0, 1, *\}^{n+k+2}$ satisfies $\mathbf{T}(\mathbf{h}) =$
 387 $*^{n+k+2}$ by the **Case (i)** and **Case (ii)**.

388 **Case (i)** : When $\mathbf{h}_{n+k+1} = 0$,

389 Eq. (3.6) can be simplified to $c_j(\mathbf{x}_{[1,n]}) \vee \mathbf{x}_{n+k+2}$. For an arbitrary element
 390 $\mathbf{z} \in v(\mathbf{h})$, we can find $j^* \in [1, k]$ such that $c_{j^*}(\mathbf{x}_{[1,n]}) = 1$ since $\forall \mathbf{y} \phi(\mathbf{y})$ is
 391 true. Consequently,

$$\begin{aligned} 392 \quad \mathbf{T}(\mathbf{h}) &\supseteq \mathbf{T}(\mathbf{h}_{[1, n+j^*-1]} \mathbf{1}_{\mathbf{h}_{[n+j^*+1, n+k+2]}}) && \because \mathbf{x}_{n+j^*} \text{ can be evaluated to be } 1 \\ 393 &\supseteq \mathbf{T}(\mathbf{h}_{[1, n+j^*-1]} \mathbf{1}_{\mathbf{h}_{[n+j^*+1, n+k]} *^2}) && \because \text{Remark 3.3} \\ 394 &\supseteq \mathbf{T}(*^{n+k+2}) && \because \text{Remark 3.4} \\ 395 &= *^{n+k+2} . \end{aligned}$$

396 **Case (ii)** : When $\mathbf{h}_{n+k+1} \in \{1, *\}$,

397 Eq. (3.8) simplifies to $\neg \mathbf{x}_{n+k+2}$. Consequently,

$$\begin{aligned} 398 \quad \mathbf{T}(\mathbf{h}) &\supseteq \mathbf{T}(\mathbf{h}_{[1, n+k+1]} *) && \because \text{Remark 3.2} \\ 399 &\supseteq \mathbf{T}(\mathbf{h}_{[1, n]} \mathbf{1}^k \mathbf{h}_{n+k+1} *) && \because \mathbf{x}_{n+k+2} \text{ can be evaluated to be } 1 \\ 400 &\supseteq \mathbf{T}(\mathbf{h}_{[1, n]} \mathbf{1}^k *^2) && \because \text{Remark 3.3} \\ 401 &\supseteq \mathbf{T}(*^{n+k+2}) && \because \text{Remark 3.4} \\ 402 &= *^{n+k+2} . \end{aligned}$$

403 Therefore, $*^{n+k+2}$ is the unique minimal trap space if $\forall \mathbf{y} \phi(\mathbf{y})$ is true. On the other
 404 hand, if $\exists \mathbf{y} \neg \phi(\mathbf{y})$ is true, then $*^{n+k+2}$ is not a minimal trap space because there is a
 405 smaller trap space $\mathbf{h}' = \mathbf{y} \mathbf{0}^{k+2}$. Hence the lemma holds. \square

406 **THEOREM 3.9.** *Given hypercube \mathbf{h} and locally-monotone BN f with local functions*
 407 *represented as propositional formulas, $\text{MINTRAP}(f, \mathbf{h})$ is coNP-hard.*

408 *Proof.* Given $n \in \mathbb{Z}^+$ and a Boolean function $\phi : \mathbb{B}^n \rightarrow \mathbb{B}$ in a DNF with k
 409 conjunctive clauses; i.e., $\phi(\mathbf{y}) := \bigvee_{j=1}^k c_j(\mathbf{y})$, consider the associated Π_1 SAT problem
 410 $\forall \mathbf{y} \phi(\mathbf{y})$, which is coNP-complete. By Lemma 3.8, this Π_1 SAT problem is true if
 411 and only if $\text{MINTRAP}(f, *^{n+k+2})$ is true for f defined by (3.5)–(3.8). Since all local
 412 functions are unate, the theorem holds. \square

413 **THEOREM 3.10.** *Given configuration \mathbf{x} and locally-monotone BN f with local*
 414 *functions represented as propositional formulas, $\text{IN-MINTRAP}(f, \mathbf{x})$ is coNP-hard.*

415 *Proof.* Suppose $n \in \mathbb{Z}^+$ and a Boolean function $\phi : \mathbb{B}^n \rightarrow \mathbb{B}$ in a DNF with
 416 k conjunctive clauses are given (i.e., $\phi(\mathbf{y}) := \bigvee_{j=1}^k c_j(\mathbf{y})$). Consider the associated
 417 Π_1 SAT problem $\forall \mathbf{y} \phi(\mathbf{y})$, which is coNP-complete. We prove the theorem by showing

418 that this $\Pi_1\text{SAT}$ problem is true if and only if $\text{IN-MINTRAP}(f, \mathbf{1}^{n+k+2})$ is true for f
 419 defined by (3.5)–(3.8).

420 If $\forall \mathbf{y} \phi(\mathbf{y})$ is true, then $*^{n+k+2}$ is the unique trap space by Lemma 3.8 and thus
 421 $\mathbf{1}^{n+k+2}$ belongs to a minimal trap space. For the remaining case where $\exists \mathbf{y} \neg\phi(\mathbf{y})$ is
 422 true,

$$\begin{aligned}
 423 \quad \mathbf{T}(\mathbf{1}^{n+k+2}) &\supseteq \mathbf{T}(\mathbf{1}^{n+k} *^2) && \because \mathbf{x}_{n+k+2} = 1 \\
 424 &\supseteq \mathbf{T}(*^{n+k+2}) && \because \text{Remark 3.4} \\
 425 &= *^{n+k+2} .
 \end{aligned}$$

426 However, $*^{n+k+2}$ is not a minimal trap space because there is a smaller trap space
 427 $\mathbf{h}' = \mathbf{y}\mathbf{0}^{k+2}$. This completes the proof. \square

428 **3.3. With local functions represented as truth tables, BDDs, and dou-**
 429 **ble DNFs.** We now consider any BN whose local functions are represented either
 430 as truth tables, BDDs, or double DNFs. In Theorem 3.13, we prove the lower bound
 431 results to the computational complexity of MINTRAP and IN-MINTRAP problems
 432 by reduction of 3DNF- $\Pi_1\text{SAT}$. Combined with the upper bound results presented in
 433 Sec. 2.3, the completeness is shown.

434 Consider the encoding of the clauses ϕ as the BN f defined by (3.5)–(3.8). Remark
 435 that all local functions but f_{n+k+1} in (3.7) depend on at most 5 variables, and thus
 436 each of them can be encoded in constant space and time as a truth table, a BDD,
 437 or double DNFs. However, the local function f_{n+k+1} in (3.7) depends on $k + 1$
 438 variables, where k is the number of clauses in the DNFs. Therefore, converting this
 439 local function may require an exponential time and space. We resolve this issue by
 440 appending a small number of auxiliary variables that correspond to local functions
 441 having a constant size. Note that (3.7) is true whenever at least one of the clauses
 442 can be evaluated to be true and the component \mathbf{x}_{n+k+2} is false. This definition can
 443 be incorporated by appending k additional components with at most two literals to
 444 the BN so that the j -th element of the new components is evaluated to be true if
 445 either c_j or the $(j - 1)$ -th element of the new components can be true for $j \in [1, k]$.
 446 As a consequence, the k -th of the additional components is true whenever at least
 447 one clause of ϕ can be evaluated to be true. We adapt this idea by expanding locally
 448 monotone BN (3.5)–(3.8) to (3.9)–(3.13), which can be encoded in constant space
 449 and time as truth table, BDD, or double DNFs. This reduction scheme is illustrated
 450 in Figure 3.5. We employ Remark 3.11 and Lemma 3.12 to prove Theorem 3.13.

$$451 \quad (3.9) \quad \forall i \in [1, n] \quad f_i(\mathbf{x}) = (\mathbf{x}_i \wedge \neg \mathbf{x}_{n+2k+1}) \vee \mathbf{x}_{n+2k+2}$$

$$452 \quad (3.10) \quad \forall j \in [1, k], \quad f_{n+j}(\mathbf{x}) = (c_j(\mathbf{x}_{[1,n]}) \wedge \neg \mathbf{x}_{n+2k+1}) \vee \mathbf{x}_{n+2k+2}$$

$$453 \quad (3.11) \quad \forall j \in [1, k], \quad f_{n+k+j}(\mathbf{x}) = \mathbf{x}_{n+j} \vee (\mathbf{x}_{n+k+j-1} \wedge (j > 1))$$

$$454 \quad (3.12) \quad f_{n+2k+1}(\mathbf{x}) = \mathbf{x}_{n+2k} \wedge \neg \mathbf{x}_{n+2k+2}$$

$$455 \quad (3.13) \quad f_{n+2k+2}(\mathbf{x}) = \mathbf{x}_{n+2k+1} \wedge \neg \mathbf{x}_{n+2k+2}$$

456

457 *Remark 3.11.* Consider a BN f given as (3.9)–(3.13). If $\mathbf{T}(\mathbf{h})_i = *$ for all
 458 $i \in [n + 1, n + k]$, we can sequentially show that $\mathbf{T}(\mathbf{h})_{n+k+j} = *$ by increasing j from
 459 1 to k . This is because $\mathbf{T}(\mathbf{h})_{n+j}$ and $\mathbf{T}(\mathbf{h})_{n+k+j-1}$ are both $*$ and thus Eq.(3.11) can
 460 be evaluated to be both 0 and 1.

461

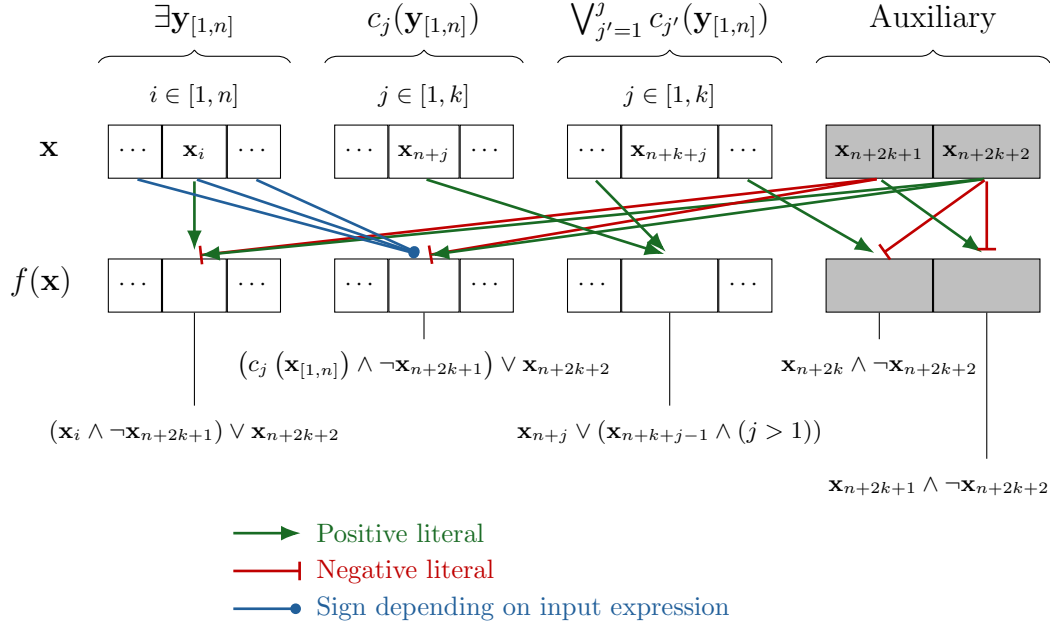


FIG. 3.5. Reduction from $\Pi_1\text{SAT}$ to MINTRAP and IN-MINTRAP for truth table, binary decision diagrams, and double DNFs (Lemma 3.12).

462 LEMMA 3.12. Consider $n \in \mathbb{Z}^+$ and a Boolean function $\phi : \mathbb{B}^n \rightarrow \mathbb{B}$ given as
 463 a 3DNF with k conjunctive clauses that contain at most three literals, i.e., $\phi(\mathbf{y}) :=$
 464 $\bigvee_{j=1}^k c_j(\mathbf{y})$. Boolean formula $\forall \mathbf{y} \phi(\mathbf{y})$ is true if and only if BN $f : \mathbb{B}^{n+2k+2} \rightarrow \mathbb{B}^{n+2k+2}$
 465 with the local functions defined by (3.9)–(3.13) has the unique minimal trap space
 466 $*^{n+2k+2}$.

467 *Proof.* If $\forall \mathbf{y} \phi(\mathbf{y})$ is true, any hypercube $\mathbf{h} \in \{0, 1, *\}^{n+2k+2}$ satisfies $\mathbf{T}(\mathbf{h}) =$
 468 $*^{n+2k+2}$ by the **Case (i)** and **Case (ii)**.

469 **Case (i)** : When $\mathbf{h}_{n+2k+1} = 0$,

470 Eq. (3.10) can be simplified to $c_j(\mathbf{x}_{[1,n]}) \vee \mathbf{x}_{n+2k+2}$ because every configuration
 471 satisfies $\mathbf{x}_{n+2k+1} = 0$. For an arbitrary element $\mathbf{z} \in v(\mathbf{h})$, we can find $j^* \in$
 472 $[1, k]$ such that $c_{j^*}(\mathbf{x}_{[1,n]}) = 1$ since $\forall \mathbf{y} \phi(\mathbf{y})$ is true. Therefore, $\mathbf{T}(\mathbf{h})_{n+j^*} \in$
 473 $\{1, *\}$ and subsequently,

$$\begin{aligned}
 474 \quad \mathbf{T}(\mathbf{h}) &\supseteq \mathbf{T}(\mathbf{h}_{[1, n+k+j^*-1]} \mathbf{1}_{[n+k+j^*+1, n+2k+2]}) \\
 475 &\quad \because \mathbf{x}_{n+k+j^*} \text{ can be evaluated to be } 1 \\
 476 &\supseteq \mathbf{T}(\mathbf{h}_{[1, n+k+j^*-1]} \mathbf{1}^{(k-j^*+1)} \mathbf{h}_{[n+2k+1, n+2k+2]}) \\
 477 &\quad \because \text{Increasing } j \text{ from } (j^* + 1) \text{ to } k, \mathbf{x}_{n+k+j} \\
 478 &\quad \text{can be sequentially evaluated to be } 1 \\
 479 &\supseteq \mathbf{T}(\mathbf{h}_{[1, n+k+j^*-1]} \mathbf{1}^{(k-j^*+1)} *^2) && \because \text{Remark 3.3} \\
 480 &\supseteq \mathbf{T}(*^{n+k} \mathbf{h}_{[n+k+1, n+k+j^*-1]} \mathbf{1}^{(k-j^*+1)} *^2) && \because \text{Remark 3.4} \\
 481 &\supseteq \mathbf{T}(*^{n+2k+2}) && \because \text{Remark 3.11} \\
 482 &= *^{n+2k+2}.
 \end{aligned}$$

483 **Case (ii)** : When $\mathbf{h}_{n+2k+1} \in \{1, *\}$,
 484 Eq. (3.13) simplifies to $\neg \mathbf{x}_{n+2k+2}$. Consequently,

$$\begin{aligned}
 485 \quad \mathbf{T}(\mathbf{h}) &\supseteq \mathbf{T}(\mathbf{h}_{[1, n+2k+1]} *) && \because \text{Remark 3.2} \\
 486 &\supseteq \mathbf{T}(\mathbf{h}_{[1, n]} \mathbf{1}^k \mathbf{h}_{[n+k+1, n+2k+1]} *) && \because \mathbf{x}_{n+2k+2} \text{ can be evaluated to be 1} \\
 487 &\supseteq \mathbf{T}(\mathbf{h}_{[1, n]} \mathbf{1}^{2k} \mathbf{h}_{n+2k+1} *) \\
 488 &\quad \because \forall j \in [1, k], \mathbf{x}_{n+k+j} && \text{can be evaluated to be 1 by } \mathbf{x}_{n+j} = 1 \\
 489 &\supseteq \mathbf{T}(\mathbf{h}_{[1, n]} \mathbf{1}^{2k} *^2) && \because \text{Remark 3.3} \\
 490 &\supseteq \mathbf{T}(*^{n+k} \mathbf{1}^k *^2) && \because \text{Remark 3.4} \\
 491 &\supseteq \mathbf{T}(*^{n+2k+2}) && \because \text{Remark 3.11} \\
 492 &= *^{n+2k+2} .
 \end{aligned}$$

493 Therefore, $*^{n+2k+2}$ is the unique minimal trap space. On the other hand, if $\exists \mathbf{y} \neg \phi(\mathbf{y})$
 494 is true, $*^{n+2k+2}$ is not a minimal trap space because there is a smaller trap space
 495 $\mathbf{h}' = \mathbf{y} \mathbf{0}^{2k+2}$. Hence the lemma holds. \square

496 **THEOREM 3.13.** *MINTRAP and IN-MINTRAP are coNP-hard for BNs with local*
 497 *functions represented with truth tables, binary decision diagrams, and double DNFs.*

498 *Proof.* The local functions defined by (3.9)–(3.13) can be encoded in a polynomial
 499 time as truth tables, BDDs, or double DNFs. Therefore, the theorem holds by Lemma
 500 3.12. \square

501 **3.4. Functional graphs of BNs .** Now consider the case when the BN $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$
 502 is represented by its functional digraph $G = (V, E)$ with $V = \mathbb{B}^n$ and
 503 $E = \{(\mathbf{x}, f(\mathbf{x})) \mid \mathbf{x} \in \mathbb{B}^n\}$. Given a vertex $\mathbf{x} \in V$, we write $out(\mathbf{x}) = \{\mathbf{y} \mid (\mathbf{x}, \mathbf{y}) \in E\}$.
 504 Note that in the case of the functional graph, $out(\mathbf{x}) = \{f(\mathbf{x})\}$, which is a singleton
 505 set. Given a set of vertices $V' \subseteq V$, we can consider a subgraph $G_{V'} = (V', \{(u, w) \in$
 506 $E \mid u \in V', w \in V'\})$.

507 For a given sub-hypercube $\mathbf{h} \in \{0, 1, *\}^n$ to be a trap space, each $\mathbf{x} \in v(\mathbf{h})$ must
 508 verify that $out(\mathbf{x}) \subseteq v(\mathbf{h})$. Therefore, TRAPSPACE can be solved in time linear to
 509 the size of G (number of vertices plus edges, $|V| + |E|$).

510 **Algorithm 3.3** for the decision of MINTRAP uses two auxiliary functions:
 511 **SUB-HYPERCUBE** (Algorithm 3.1) and **SATURATE** (Algorithm 3.2). The function
 512 **SUB-HYPERCUBE** returns the smallest enclosing sub-hypercube for a given a non-empty
 513 sublist of vertices $W \subseteq V$. From the resulting sub-hypercube, the function **SATURATE**
 514 computes its smallest enclosing trap space.

Algorithm 3.1 SUB-HYPERCUBE

Input: Sublist of vertices W

Output: The smallest sub-hypercube enclosing W ($:= \mathbf{h}$)

```

1:  $\mathbf{h} = W_1$ 
2: for  $\mathbf{x} \in W$  do
3:   for  $i \in [1, n]$  do
4:     if  $(\mathbf{h}_i \in \mathbb{B})$  and  $(\mathbf{h}_i \neq \mathbf{x}_i)$  then
5:        $\mathbf{h}_i := *$ 
6: return  $\mathbf{h}$ 
    
```

Algorithm 3.2 SATURATE

Input: Sublist of vertices W **Output:** The smallest sub-hypercube enclosing W closed by f ($:= \mathbf{h}$)

```

1:  $\mathbf{h} := \text{SUB-HYPERCUBE}(W)$ 
2: repeat
3:    $\mathbf{h}' := \mathbf{h}$ 
4:    $W := v(\mathbf{h}) \cup \bigcup_{u \in v(\mathbf{h})} \text{out}(u)$ 
5:    $\mathbf{h} := \text{SUB-HYPERCUBE}(W)$ 
6: until  $\mathbf{h} = \mathbf{h}'$ 
7: return  $\mathbf{h}$ 

```

515 Remark that SATURATE runs in a polynomial time to the size of G as the loop in
516 line 2-5 is performed at most n times.

517 One can decide whether the sub-hypercube \mathbf{h} is a minimal trap space by com-
518 puting the terminal strongly connected components of G which are enclosed in \mathbf{h}
519 and verify that their smallest enclosing trap space is \mathbf{h} . Indeed, consider that \mathbf{h} is a
520 trap space. By definition, the saturation of any set of its vertices gives a trap spaces
521 which is either equal to or smaller than \mathbf{h} . Then, remark that any trap space within
522 \mathbf{h} contains at least one terminal strongly connected component of $G_{v(\mathbf{h})}$. Therefore,
523 it is sufficient to verify that the saturation of all these terminal strongly connected
524 components are not strictly smaller than \mathbf{h} to determine that \mathbf{h} is minimal.

525 We call the algorithm computing the terminal strongly connected components
526 **terminal-SCCs** and it can be done in a polynomial time to the size of G (e.g., with
527 Tarjan's algorithm [33]).

Algorithm 3.3 MINTRAP (functional graph)

Input: The BN G , a candidate minimal trap space \mathbf{h} **Output:** Whether \mathbf{h} is a minimal trap space of G

```

1: if not TRAPSPACE( $G, \mathbf{h}$ ) then
2:   return False
3: tSCCs := terminal-SCCs( $G_{v(\mathbf{h})}$ )
4: for each  $W$  in tSCCs do
5:   if SATURATE( $W$ )  $\neq \mathbf{h}$  then
6:     return False
7: return True

```

528 This algorithm runs in a polynomial time to the size of G . Finally, remark that
529 $\text{IN-MINTRAP}(f, \mathbf{x})$ can be decided using $\text{IS_MINTRAP}(G, \text{SATURATE}(\{\mathbf{x}\}))$, which also
530 runs in a polynomial time to the size of G .

531 **THEOREM 3.14.** *TRAPSPACE, MINTRAP, and IN-MINTRAP are in P for BNs*
532 *given as their functional graph.*

533 The functional graph of f corresponds to the so-called state transition graph with
534 the synchronous (parallel) update mode: each edge corresponds to a synchronous
535 transition. One can remark that the above algorithms give equivalent results with the
536 fully asynchronous state transition graph where $\text{out}'(\mathbf{x}) = \{\mathbf{y} \in \mathbb{B}^n \mid \exists i \in [1, n], \mathbf{y}_i =$
537 $f_i(\mathbf{x}), \forall j \in [1, n], j \neq i, \mathbf{x}_j = \mathbf{y}_j\}$. Indeed, $\text{SUB-HYPERCUBE}(\{\mathbf{x}, f(\mathbf{x})\})$ is always equal
538 to $\text{SUB-HYPERCUBE}(\{\mathbf{x}\} \cup \text{out}'(\mathbf{x}))$; remark that, for any $i \in [1, n]$, $f_i(\mathbf{x}) \neq \mathbf{x}_i$ if and

539 only if there exists $\mathbf{y} \in out'(\mathbf{x})$ such that $\mathbf{y}_i \neq \mathbf{x}_i$.

540 **4. Conclusion.** In this paper, we characterized the computational complexity
 541 of three important decision problems related to trap spaces in Boolean networks con-
 542 sidering various representations and the locally monotone case. We demonstrated
 543 that, in general, determining minimal trap space properties and the membership of
 544 configurations to minimal trap spaces are equivalent to solving the satisfiability of
 545 Boolean formulas with two alternating quantifiers \forall and \exists . Hence, our results show
 546 that they are Π_2^P -complete. However, whenever restricting to the cases whenever BN
 547 is locally monotone, or whenever its local functions are encoded as truth tables, bi-
 548 nary decision diagrams, or double DNFs (such as Petri nets encodings of BNs), the
 549 complexity drops by one level in the polynomial hierarchy and becomes equivalent to
 550 Π_1 SAT. These three latter encodings are well-known representations for which SAT
 551 and Π_1 SAT decisions are in P. Future work may consider other encodings sharing these
 552 complexity properties, such as deterministic decomposable negation normal forms [9],
 553 and deriving more generic proofs to this class of encodings. Finally, whenever the
 554 BN is given by its functional graph (corresponding to its synchronous state transition
 555 graph), minimal trap space properties can be decided by deterministic algorithms in
 556 a polynomial time.

557 In practice, solving coNP problems can be tackled with SAT solvers, whereas
 558 solving Π_2^P necessitates more elaborated approaches, such as Answer-Set Program-
 559 ming [11]. Another recent approach is to decompose the problem into two parts and
 560 alternately solving them; one seeks for a candidate solution by relaxing quantifier \forall as
 561 \exists , and then another verifies whether the candidate is valid to the original problem. If
 562 not, a proper constraint is added to the prior problem to remove the candidate from
 563 the solution space, and the procedure is repeated. This approach has been used to
 564 control minimal trap spaces [29] or fixed points [19].

565 Future direction may consider studying the computational complexity of problems
 566 related to the set of minimal trap spaces of a BN, such as deciding whether all the
 567 minimal trap spaces satisfy a given property. This will give insight into the complexity
 568 for control problems related to minimal trap spaces in BNs, as tackled in [22, 30].

569

REFERENCES

- 570 [1] T. AKUTSU, *Algorithms for Analysis, Inference, and Control of Boolean Networks*, World
 571 Scientific, 2018.
- 572 [2] J. ARACENA, E. FANCHON, M. MONTALVA, AND M. NOUAL, *Combinatorics on update digraphs*
 573 *in Boolean networks*, Discrete Applied Mathematics, 159 (2011), pp. 401–409, <https://doi.org/10.1016/j.dam.2010.10.010>.
- 574 [3] J. ARACENA, E. GOLES, A. MOREIRA, AND L. SALINAS, *On the robustness of update schedules in*
 575 *Boolean networks*, Biosystems, 97 (2009), pp. 1 – 8, <https://doi.org/10.1016/j.biosystems.2009.03.006>.
- 576 [4] S. ARORA AND B. BARAK, *Computational Complexity: A Modern Approach*, Cambridge Uni-
 577 versity Press, Cambridge, 2009.
- 578 [5] F. BRIDOUX, M. GADOLEAU, AND G. THEYSSIER, *On simulation in automata networks*, in
 579 *Lecture Notes in Computer Science*, Springer International Publishing, 2020, pp. 277–288,
 580 https://doi.org/10.1007/978-3-030-51466-2_24.
- 581 [6] C. CHAOUÏYA, A. NALDI, E. REMY, AND D. THIEFFRY, *Petri net representation of multi-valued*
 582 *logical regulatory graphs*, Natural Computing, 10 (2010), pp. 727–750, <https://doi.org/10.1007/s11047-010-9178-0>.
- 583 [7] T. CHATAIN, S. HAAR, L. JEZEQUEL, L. PAULEVÉ, AND S. SCHWOON, *Characterization of reach-*
 584 *able attractors using Petri net unfoldings*, in *Computational Methods in Systems Biology*,
 585 vol. 8859 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Cham, 2014,
 586 pp. 129–142, https://doi.org/10.1007/978-3-319-12982-2_10.

589

- 590 [8] Y. CRAMA AND P. L. HAMMER, *Boolean Functions: Theory, Algorithms, and Applications*,
 591 no. 142 in *Encyclopedia of Mathematics and Its Applications*, Cambridge University Press,
 592 Cambridge ; New York, 2011.
- 593 [9] A. DARWICHE, *On the tractable counting of theory models and its application to truth main-*
 594 *tenance and belief revision*, *Journal of Applied Non-Classical Logics*, 11 (2001), p. 11–34,
 595 <https://doi.org/10.3166/jancl.11.11-34>, <http://dx.doi.org/10.3166/jancl.11.11-34>.
- 596 [10] R. DRECHSLER AND B. BECKER, *Binary Decision Diagrams*, Springer US, Boston, MA, 1998.
- 597 [11] T. EITER AND G. GOTTLÖB, *On the computational cost of disjunctive logic programming:*
 598 *Propositional case*, *Annals of Mathematics and Artificial Intelligence*, 15 (1995), pp. 289–
 599 323, <https://doi.org/10.1007/bf01536399>.
- 600 [12] G. GAMARD, P. GUILLON, K. PERROT, AND G. THEYSSIER, *Rice-like theorems for automata*
 601 *networks*, in 38th International Symposium on Theoretical Aspects of Computer Science
 602 (STACS 2021), M. Bläser and B. Monmege, eds., vol. 187 of *Leibniz International Proceed-*
 603 *ings in Informatics (LIPIcs)*, Dagstuhl, Germany, 2021, Schloss Dagstuhl – Leibniz-Zentrum
 604 für Informatik, pp. 32:1–32:17, <https://doi.org/10.4230/LIPIcs.STACS.2021.32>.
- 605 [13] M. GRABISCH AND A. RUSINOWSKA, *A model of influence based on aggregation functions*, *Math-*
 606 *ematical Social Sciences*, 66 (2013), pp. 316–330, <https://doi.org/10.1016/j.mathsocsci.2013.07.003>.
- 608 [14] S. A. KAUFFMAN, *Metabolic stability and epigenesis in randomly connected nets*, *Journal*
 609 *of Theoretical Biology*, 22 (1969), pp. 437–467, [https://doi.org/10.1016/0022-5193\(69\)](https://doi.org/10.1016/0022-5193(69)90015-0)
 610 [90015-0](https://doi.org/10.1016/0022-5193(69)90015-0).
- 611 [15] S. A. KAUFFMAN, *The origins of order: self-organization and selection in evolution*, Oxford
 612 University Press, USA, 1993.
- 613 [16] H. KLARNER, A. BOCKMAYR, AND H. SIEBERT, *Computing maximal and minimal trap spaces of*
 614 *Boolean networks*, *Natural Computing*, 14 (2015), pp. 535–544, [https://doi.org/10.1007/](https://doi.org/10.1007/s11047-015-9520-7)
 615 [s11047-015-9520-7](https://doi.org/10.1007/s11047-015-9520-7).
- 616 [17] H. KLARNER AND H. SIEBERT, *Approximating attractors of Boolean networks by iterative CTL*
 617 *model checking*, *Frontiers in Bioengineering and Biotechnology*, 3 (2015), [https://doi.org/](https://doi.org/10.3389/fbioe.2015.00130)
 618 [10.3389/fbioe.2015.00130](https://doi.org/10.3389/fbioe.2015.00130).
- 619 [18] A. MONTAGUD, J. BÉAL, L. TOBALINA, P. TRAYNARD, V. SUBRAMANIAN, B. SZALAI,
 620 R. ALFÖLDI, L. PUSKÁS, A. VALENCIA, E. BARILLOT, J. SAEZ-RODRIGUEZ, AND L. CAL-
 621 ZONE, *Patient-specific Boolean models of signalling networks guide personalised treatments*,
 622 *eLife*, 11 (2022), <https://doi.org/10.7554/elife.72626>.
- 623 [19] K. MOON, K. LEE, S. CHOPRA, AND S. KWON, *Bilevel integer programming on a Boolean*
 624 *network for discovering critical genetic alterations in cancer development and therapy*,
 625 *European Journal of Operational Research*, 300 (2022), pp. 743–754, [https://doi.org/10.](https://doi.org/10.1016/j.ejor.2021.10.019)
 626 [1016/j.ejor.2021.10.019](https://doi.org/10.1016/j.ejor.2021.10.019).
- 627 [20] A. NALDI, C. HERNANDEZ, W. ABOU-JAOUDE, P. T. MONTEIRO, C. CHAOUIYA, AND D. THI-
 628 EFFRY, *Logical modeling and analysis of cellular regulatory networks with GINsim 3.0*,
 629 *Frontiers in Physiology*, 9 (2018), <https://doi.org/10.3389/fphys.2018.00646>.
- 630 [21] L. PAULEVÉ, *Reduction of qualitative models of biological networks for transient dynamics*
 631 *analysis*, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15
 632 (2018), pp. 1167–1179, <https://doi.org/10.1109/TCBB.2017.2749225>.
- 633 [22] L. PAULEVÉ, *Marker and source-marker reprogramming of Most Permissive Boolean networks*
 634 *and ensembles with BoNesis*, 2022, <https://doi.org/10.48550/ARXIV.2207.13307>.
- 635 [23] L. PAULEVÉ, J. KOLČÁK, T. CHATAIN, AND S. HAAR, *Reconciling qualitative, abstract, and*
 636 *scalable modeling of biological networks*, *Nature Communications*, 11 (2020), p. 4256, [https://](https://doi.org/10.1038/s41467-020-18112-5)
 637 doi.org/10.1038/s41467-020-18112-5.
- 638 [24] L. PAULEVÉ AND A. RICHARD, *Static analysis of Boolean networks based on interaction graphs:*
 639 *A survey*, *Electronic Notes in Theoretical Computer Science*, 284 (2012), pp. 93–104, [https://](https://doi.org/10.1016/j.entcs.2012.05.017)
 640 doi.org/10.1016/j.entcs.2012.05.017.
- 641 [25] L. PAULEVÉ AND S. SENÉ, *Non-deterministic updates of Boolean networks*, in 27th IFIP
 642 WG 1.5 International Workshop on Cellular Automata and Discrete Complex Systems
 643 (AUTOMATA 2021), vol. 90 of *Open Access Series in Informatics (OASIcs)*, Dagstuhl,
 644 Germany, 2021, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 10:1–10:16,
 645 <https://doi.org/10.4230/OASIcs.AUTOMATA.2021.10>.
- 646 [26] L. PAULEVÉ AND S. SENÉ, *Boolean networks and their dynamics: the impact of updates*, in
 647 *Systems Biology Modelling and Analysis: Formal Bioinformatics Methods and Tools*, Wi-
 648 ley, 2022. In press. Preprint available at [https://pageperso.lis-lab.fr/~sylvain.sene/files/](https://pageperso.lis-lab.fr/~sylvain.sene/files/publi_pres/ps22.pdf)
 649 [publi_pres/ps22.pdf](https://pageperso.lis-lab.fr/~sylvain.sene/files/publi_pres/ps22.pdf).
- 650 [27] A. POINDRON, *A general model of binary opinions updating*, *Mathematical Social Sciences*, 109
 651 (2021), pp. 52–76, <https://doi.org/10.1016/j.mathsocsci.2020.10.004>.

- 652 [28] A. RICHARD, *Positive circuits and maximal number of fixed points in discrete dynamical sys-*
653 *tems*, Discrete Applied Mathematics, 157 (2009), pp. 3281–3288, [https://doi.org/10.1016/](https://doi.org/10.1016/j.dam.2009.06.017)
654 [j.dam.2009.06.017](https://doi.org/10.1016/j.dam.2009.06.017).
- 655 [29] S. RIVA, J.-M. LAGNIEZ, G. M. LÓPEZ, AND L. PAULEVÉ, *Tackling Universal Properties of Min-*
656 *imal Trap Spaces of Boolean Networks*, in Computational Methods in Systems Biology,
657 J. Pang and J. Niehren, eds., Lecture Notes in Computer Science, Cham, 2023, Springer
658 Nature Switzerland, pp. 157–174.
- 659 [30] J. C. ROZUM, D. DERITEI, K. H. PARK, J. G. T. ZAÑUDO, AND R. ALBERT, *pystablemotifs:*
660 *Python library for attractor identification and control in Boolean networks*, Bioinformatics,
661 38 (2021), pp. 1465–1466, <https://doi.org/10.1093/bioinformatics/btab825>.
- 662 [31] J. D. SCHWAB, S. D. KÜHLWEIN, N. IKONOMI, M. KÜHL, AND H. A. KESTLER, *Concepts in Bool-*
663 *ean network modeling: What do they all mean?*, Computational and Structural Biotech-
664 nology Journal, 18 (2020), pp. 571–582, <https://doi.org/10.1016/j.csbj.2020.03.001>.
- 665 [32] L. J. STOCKMEYER, *The polynomial-time hierarchy*, Theoretical Computer Science, 3 (1976),
666 pp. 1–22, [https://doi.org/10.1016/0304-3975\(76\)90061-X](https://doi.org/10.1016/0304-3975(76)90061-X).
- 667 [33] R. TARJAN, *Depth-first search and linear graph algorithms*, SIAM Journal on Computing, 1
668 (1972), pp. 146–160, <https://doi.org/10.1137/0201010>.
- 669 [34] R. THOMAS, *Boolean formalization of genetic control circuits*, Journal of Theoretical Biology,
670 42 (1973), pp. 563 – 585, [https://doi.org/10.1016/0022-5193\(73\)90247-6](https://doi.org/10.1016/0022-5193(73)90247-6).
- 671 [35] V.-G. TRINH, B. BENHAMOU, K. HIRAISHI, AND S. SOLIMAN, *Minimal trap spaces of Logical*
672 *models are maximal siphons of their Petri net encoding*, in CMSB 2022 - International
673 Conference on Computational Methods in Systems Biology, 2022, [https://doi.org/10.1007/](https://doi.org/10.1007/978-3-031-15034-0_8)
674 [978-3-031-15034-0_8](https://doi.org/10.1007/978-3-031-15034-0_8).
- 675 [36] I. WEGENER, *BDDs—design, analysis, complexity, and applications*, Discrete Applied Mathe-
676 matics, 138 (2004), pp. 229–251, [https://doi.org/10.1016/s0166-218x\(03\)00297-x](https://doi.org/10.1016/s0166-218x(03)00297-x).
- 677 [37] J. G. T. ZAÑUDO, P. MAO, C. ALCON, K. KOWALSKI, G. N. JOHNSON, G. XU, J. BASELGA,
678 M. SCALTRITI, A. LETAI, J. MONTERO, R. ALBERT, AND N. WAGLE, *Cell line-specific*
679 *network models of ER+ breast cancer identify potential PI3K α inhibitor resistance mech-*
680 *anisms and drug combinations*, Cancer Research, 81 (2021), pp. 4603–4617, [https://doi.](https://doi.org/10.1158/0008-5472.can-21-1208)
681 [org/10.1158/0008-5472.can-21-1208](https://doi.org/10.1158/0008-5472.can-21-1208).