Decomposability of partially defined Boolean functions

Endre Boros*, Vladimir Gurvicha.1, Peter L. Hammera.1, Toshihide Ibarakib, Alexander Koganb

*aRUTCOR, Rutgers University, P.O. Box 5062, New Brunswick, NJ 08903, USA
bDepartment of Applied Mathematics and Physics, Faculty of Engineering, Kyoto University, Kyoto 606, Japan

Received 28 March 1994; revised 12 October 1994

Abstract

The problem of recognizing decomposability of partially defined Boolean functions is considered. The results include polynomial time algorithms for certain important types of decomposition, as well as NP-completeness proofs for more complex structures.

1. Introduction

A typical mathematical problem frequently arising in numerous areas (e.g., artificial intelligence, learning theory, game theory, reliability theory, VLSI design, combinatorial optimization, etc.) will be stated below, along with a brief indication of its particular meaning in some of these areas.

Given a set of data, represented as a set of binary “true n-vectors” (or “positive examples”) and a disjoint set of “false n-vectors” (or “negative examples”), along with a family $S_0, S_1, \ldots, S_k$ of subsets of $\{1, 2, \ldots, n\}$, we have to establish the existence of Boolean functions $g, h_1, \ldots, h_k$, if any, so that $g(S_0, h_1(S_1), \ldots, h_k(S_k))$ is true (false) in every given true (false) vector.

For instance, in the specific case of artificial intelligence, the given vectors represent positive and negative examples, while their components correspond to attributes. In this case, the sets $S_i$ represent groups of attributes. The question is whether these...
interactive groups define new "meta-attributes", which can completely specify the positive or negative character of the examples. This problem of structure identification is in fact one form of knowledge acquisition – an important topic in artificial intelligence.

As an illustration, let us assume that the components of the given binary vectors are associated with a list of food items, and that the positive and negative example vectors represent observations on the days when a patient had or did not have a headache, while the sets $S_1$ and $S_2$, respectively, denote the (possibly overlapping) sets of food items containing proteins and carbohydrates; let $S_0$ represent the set of food items low in both proteins and carbohydrates. In order to test the hypothesis that insufficient variety in the types of proteins and in the types of carbohydrates during the day leads to headaches, we have to establish the existence or absence of a Boolean function $g(S_0, h_1(S_1), h_2(S_2))$ which separates correctly the positive and negative examples.

In addition to knowledge acquisition, such decompositions can be encountered in various applications such as learning theory (where the process of theory formation can be greatly enhanced by the knowledge of the underlying structure that holds among the relevant attributes [7,17]), game theory (where the sets $S_i$ represent coalitions of players [18,21]), reliability theory (where $g(S_0, h_1(S_1), \ldots, h_k(S_k))$ represents the decomposition structure of the system under consideration [13,18,21]), relational databases (where decompositions not only save storage but also speed up future queries (see e.g., [10,15])), VLSI design (where decompositions can help both minimization and testability (see e.g., [6,8,12,16])), combinatorial optimization (where the recognition of hierarchical structures is useful in designing efficient algorithms [14]), etc.

To formulate the problem precisely, we introduce the notion of a partially defined Boolean function (pdBf), which is defined as a pair of disjoint sets $(T, F)$ of binary $n$-vectors. The set $T$ denotes the set of true vectors (or positive examples) and $F$ denotes the set of false vectors (or negative examples). A Boolean function $f: \{0,1\}^n \rightarrow \{0,1\}$ is called an extension (or a theory) of the pdBf $(T, F)$ if $f(x) = 1$ for all $x \in T$ and $f(y) = 0$ for all $y \in F$.

Evidently, the disjointness of the sets $T$ and $F$ is a necessary and sufficient condition for the existence of an extension. It may not be evident, however, to find out whether a given pdBf has an extension with a specified decomposition structure. There is a vast literature on the decomposition of completely specified Boolean functions (see [1,3,9,19,20]), with particular attention given to the monotone case (see [18]). Unfortunately, these results cannot be applied directly to partially defined Boolean functions. Typically, a pdBf may have exponentially many different extensions, many of which may have exponential sizes. The level of decomposability as well as the complexity of its recognition may vary widely for these extensions.

In this paper we formulate the problem of deciding the existence of an extension with a specified decomposition structure (without the need of a complete description of such an extension), and study its computational complexity. We obtain computationally efficient algorithms in some cases, and prove NP-completeness in some other cases.
2. Main results

In this section we shall state the main results, leaving the technical details and proofs for the following sections.

Let $V = \{x_1, x_2, \ldots, x_n\}$ denote the set of Boolean variables (attributes, or components). For a Boolean vector $x$ and for a subset $S \subseteq V$ of the variables, let $x[S]$ denote the projection of $x$ on $S$, i.e., the vector restricted to the variables in $S$. For instance, if $x = (1011100)$, $y = (0010011)$, and $S = \{x_2, x_3, x_5\}$, then $x[S] = (011)$, and $y[S] = (010)$. To simplify notation, for a Boolean function $h$ depending only on variables of $S \subseteq V$, we write simply $h(S)$ instead of $h(x[S])$. Furthermore, if $S$ is a singleton, say $S = \{x_j\}$, and $h$ depends only on $S$, then we can assume without loss of generality that $h(S) = x_j$.

Now we are ready to give a precise formulation of our problem.

Given a pdBF $(T, F)$ and a family of subsets $\mathcal{S} = \{S_i | S_i \subseteq V, i = 0, \ldots, k\}$, the decomposability problem is the problem of deciding the existence of an extension $f$ of $(T, F)$ for which there exist Boolean functions $h_1, \ldots, h_k$, and $g$ satisfying the following conditions:

(i) $h_i$ depends only on variables in $S_i$, $i = 1, \ldots, k$,

(ii) $g$ depends on the variables in $S_0$ and on the binary values $h_i(S_i)$ for $i = 1, \ldots, k$ (i.e. $g: \{0, 1\}^{|S_0| + k} \rightarrow \{0, 1\}$),

(iii) $f = g(S_0, h_1(S_1), h_2(S_2), \ldots, h_k(S_k))$.

The decomposition (1) associated to the given family of subsets $\mathcal{S}$ will be called a scheme. Let us note that $S_0, S_1, \ldots, S_k$ are not assumed to be disjoint, and that it is also possible that some of these subsets are equal, e.g. $S_1 = S_2$. To further simplify notations, if $S_0 = \emptyset$, we omit $S_0$ from formula (1), i.e., we write $f = g(h_1(S_1), h_2(S_2), \ldots, h_k(S_k))$.

In these terms, our objective is to find out whether a given pdBF has an extension of a given scheme, i.e. whether it has a representation of the type (1). It is also important to establish whether this question can be answered in time which is polynomial in $n$ and $m = |T| + |F|$.

Let us state now the main result of this paper for the case of general extensions.

**Theorem 1.** Let $(T, F)$ be a given pdBF, $n = |V|$ and $m = |T| + |F|$. Then the existence of extensions of the following schemes

(a) $f = g(S_0, h_1(S_1))$,

(b) $f = g(h_1(S_1), h_2(S_2))$,

(c) $f = g(S_0, h_1(S_1), h_2(S_2))$ and $|S_0| = O(\max(\log \log n, \log \log m))$

can be decided in time polynomial in $n$ and $m$. However, deciding the existence of extensions of the schemes

(d) $f = g(S_0, h_1(S_1), h_2(S_2))$,

(e) $f = g(h_1(S_1), h_2(S_2), h_3(S_3))$

are NP-complete problems.
Obviously, a scheme of type (b) is a special case of schemes of type (c). Note however that there may exist more efficient algorithms for case (b) than for case (c). Therefore, in the subsequent sections, we give complete proofs for all of the above five cases.

Let us remark that it may be important and interesting to consider additional restrictions on the properties of the functions $g$ and $h_i (i \in \{1, \ldots, k\})$, e.g., one may require these functions to be positive, Horn, etc. In particular, if all the functions $h_1, \ldots, h_k$ and $g$ are required to be positive, we shall call the specification (1) a positive scheme. A Boolean function $f$ is called positive (or monotone) if $u \leq w$ (componentwise) always implies $f(u) \leq f(w)$. Since positive functions constitute a frequently studied case of Boolean functions, we pay special attention to this class.

**Theorem 2.** Let $(T, F)$ be a given pdBf, $n = \lvert V \rvert$ and $m = \lvert T \rvert + \lvert F \rvert$. Then the existence of extensions of the following positive schemes

(a) $f = g(S_0, h_1(S_1))$,
(b) $f = g(h_1(S_1), h_2(S_2))$,
(c) $f = g(S_0, h_1(S_1), h_2(S_2))$,
(d) $f = g(h_1(S_1), h_2(S_2), h_3(S_3))$

can be decided in time polynomial in $n$ and $m$.

Let us note that, although (c) and (d) include (a) and (b) as special cases, we can get more efficient algorithms for the simpler cases. Therefore we give separate proofs for all of the above four cases in the second half of this paper.

3. Basic notations for the general case

To a given pdBf $(T, F)$ and a given scheme $f = g(S_0, h_1(S_1), \ldots, h_k(S_k))$, we associate its structure hypergraph $H = (V, E)$ defined as follows:

$$V = V_0 \cup V_1 \cup \cdots \cup V_k \quad \text{and} \quad E = E_1 \cup E_0,$$

where

- $V_i = \{x[S_i] \mid x \in T \cup F\}$, $i = 0, 1, \ldots, k$,
- $E_1 = \{(x[S_0], x[S_1], \ldots, x[S_k]) \mid x \in T\}$,
- $E_0 = \{(x[S_0], x[S_1], \ldots, x[S_k]) \mid x \in F\}$.

The hyperedges $e = (v_0, \ldots, v_k) \in E_1$ and $e' = (v'_0, \ldots, v'_k) \in E_0$ will be referred simply as true-edges and false-edges, respectively. In pictures, true-edges will be drawn as solid lines, and false-edges will be printed as dotted lines. If necessary, the vectors $x$ are
indicated besides the corresponding hyperedges. Note that $H$ has at most $(k + 1)n$ vertices and exactly $m$ hyperedges, where $m = |T| + |F|$.

**Example 1.** Let us consider the pdBf given in the truth table of Fig. 1, and the scheme $f = g(S_0, h_1(S_1))$, where $S_0 = \{x_1, x_2, x_3\}$ and $S_1 = \{x_4, x_5, x_6\}$. The corresponding structure hypergraph (which is a graph in this case) is also given in Fig. 1.

The following proposition characterizing the existence of an extension of a given scheme for a given pdBf follows immediately from the definition of scheme and structure hypergraph.

**Proposition 1.** A pdBf $(T, F)$ has an extension of scheme $f = g(S_0, h_1(S_1), \ldots, h_k(S_k))$ if and only if there are binary assignments $h_i: V_i \rightarrow \{0, 1\}$, $i = 1, 2, \ldots, k$, such that no pair of true- and false-edges receive the same bit pattern, i.e. if and only if for every pair consisting of a true-edge $e = (v_0, v_1, \ldots, v_k) \in E_1$ and a false-edge $e' = (v'_0, v'_1, \ldots, v'_k) \in E_0$, either $v_0 \neq v'_0$ or there exists an index $i$ for which $h_i(v_i) \neq h_i(v'_i)$.

For example, the assignment $h_1: V_1 \rightarrow \{0, 1\}$ indicated in the structure graph $H$ of Fig. 1 obviously satisfies the condition in this proposition. Therefore, the pdBf of Example 1 has an extension of scheme $f = g(S_0, h_1(S_1))$. In the subsequent sections, we investigate the complexity of finding such assignments $h_i$, $i = 1, 2, \ldots, k$, for some selected types of schemes.

\[
\begin{array}{c|cc|cc}
 & S_0 & \quad & S_1 & \quad \\
\hline
T & 1 & 0 & 0 & 1 & 0 & 1 \\
 & 0 & 1 & 1 & 1 & 1 & 0 \\
F & 0 & 1 & 1 & 0 & 1 & 0 \\
 & 1 & 1 & 0 & 1 & 0 & 1 \\
 & 1 & 0 & 0 & 1 & 1 & 0 \\
 & 0 & 0 & 0 & 1 & 1 & 0 \\
 & 0 & 0 & 0 & 0 & 1 & 0 \\
\end{array}
\]

\[
H
\]

\[
\begin{array}{c|c}
 & V_0 & V_1 \\
\hline
V_0 & 100 & 101 \\
 & 011 & 110 \\
 & 110 & 010 \\
 & 000 & 1 \\
\end{array}
\]

Fig. 1. A pdBf and its structure hypergraph.
4. Scheme \( f = g(S_0, h_1(S_1)) \)

To a given pdBf \((T, F)\), let us associate its conflict graph \( G = (V, E) \) defined by

\[
V = \{x[S_j] | x \in T \cup F\},
\]

\[
E = \{(x[S_j], y[S_j]) | x \in T, y \in F, x[S_0] = y[S_0]\}.
\]

For instance, the conflict graph of the pdBf of Example 1 is shown in Fig. 2.

**Proposition 2.** A pdBf \((T, F)\) has an extension of scheme \( f = g(S_0, h_1(S_1)) \) if and only if its conflict graph \( G \) is bipartite.

**Proof.** If \( G = (V, E) \) is bipartite, then let \( A \) and \( B \) be the partition of \( V \) such that any edge in \( E \) has one of its vertices in \( A \) and the other in \( B \). We can now define \( h_1 \) by assigning 0 to all the vertices in \( A \), and assigning 1 to all the vertices in \( B \). Then, by the definition of \( G \), there cannot be a pair \( x \in T, y \in F \) of true and false vectors, such that the assignments \((x[S_0], h_1(x[S_1]))\) and \((y[S_0], h_1(y[S_1]))\) coincide. Therefore, the given pdBf has an extension of the required scheme, by Proposition 1.

Conversely, if the pdBf \((T, F)\) has an extension of scheme \( f = g(S_0, h_1(S_1)) \), then the values of \( h_1 \) must define a bipartition of \( V \). \( \square \)

Since constructing the graph \( G \) and checking its bipartiteness can be done in polynomial time, it follows immediately that:

**Proposition 3.** Given a pdBf \((T, F)\) the existence of an extension of scheme \( f = g(S_0, h_1(S_1)) \) can be checked in polynomial time.
5. Scheme \( f = g(h_1(S_1), h_2(S_2)) \)

Since no restrictions are imposed on the functions \( g, h_1, \) and \( h_2 \) of scheme \( f = g(h_1(S_1), h_2(S_2)) \), the functional form of \( g \) can be assumed, without loss of generality, to be one of the following three types:

\[
g = h_1 \land h_2 = h_1 h_2 \quad \text{(conjunction),}
\]
\[
g = h_1 \lor h_2 \quad \text{(disjunction),}
\]
\[
g = h_1 \land \neg h_2 \lor \neg h_1 \land h_2 \quad \text{(exclusive-or).}
\]

To simplify arguments in the sequel, we study the conditions for the existence of extensions of these types separately. We note here that the structure hypergraph \( H = (V, E) \) of scheme \( f = g(h_1(S_1), h_2(S_2)) \) is always a bipartite graph, since \( V \) is partitioned into \( V_1 \) and \( V_2 \), and all edges \((a, b)\) satisfy \( a \in V_1 \) and \( b \in V_2 \).

5.1. Conjunction \( g = h_1 \land h_2 \)

**Proposition 4.** An extension of type \( g = h_1 \land h_2 \) exists if and only if the structure hypergraph \( H \) does not contain the subgraph of Fig. 3, i.e. if and only if \( H \) has no four vertices \( a, b, c, d \) such that \( a, c \in V_1, b, d \in V_2, (a, b), (c, d) \in E_1 \) and \( (c, b) \in E_0 \).

**Proof.** It is immediate from Proposition 1 and the definition of \( H \) that this extension is possible if and only if \( V_1 \) and \( V_2 \) have binary assignments \( h_1 \) and \( h_2 \) such that

\[
(a, b) \in E_1 \Rightarrow h_1(a) = h_2(b) = 1
\]

and

\[
(a, b) \in E_0 \Rightarrow \text{either } h_1(a) = 0 \text{ or } h_2(b) = 0 \text{ or both.}
\]

Therefore, if \( H \) has four vertices \( a, b, c, d \) such that \( a, c \in V_1, b, d \in V_2, (a, b), (c, d) \in E_1 \) and \( (c, b) \in E_0 \), then \( (a, b), (c, d) \in E_1 \) imply that \( h_1(c) = h_2(b) = 1 \), which is impossible since \( (c, b) \in E_0 \).

Conversely, if \( H \) has no four vertices \( a, b, c, d \) such that \( a, c \in V_1, b, d \in V_2, (a, b), (c, d) \in E_1 \) and \( (c, b) \in E_0 \), then assign 1 to all vertices incident with edges of \( E_1 \), and assign 0 otherwise. Then both \( V_1 \) and \( V_2 \) will be partitioned as \( V_1 = V_1^1 \cup V_1^0 \),

![Fig. 3. The forbidden subgraph of Proposition 4.](image-url)
Example 2. Consider the pdBf \((T, F)\) and its structure (hyper) graph \(H\) of Fig. 4. This pdBf does not have an extension of type \(g = h_1 \wedge h_2\), since \(H\) contains a subgraph of Proposition 4, i.e. vertices \((110), (101) \in V_1\), \((100), (010) \in V_2\), and edges \(((110), (100)), ((101), (010)) \in E_1\), \(((101), (100)) \in E_0\).

5.2. Disjunction \(g = h_1 \lor h_2\)

Proposition 5. An extension of type \(g = h_1 \lor h_2\) exists if and only if the structure hypergraph \(H\) does not contain the subgraph of Fig. 5, i.e. if and only if the graph \(H\) has no four vertices \(a, b, c, d\) such that \(a, c \in V_1\), \(b, d \in V_2\), \((a, b), (c, d) \in E_0\) and \((c, b) \in E_1\).
Proof. Apply an argument dual to the proof of Proposition 4.

The pdBf of Example 2 does not have an extension of this type either, since its structure hypergraph $H$ contains the subgraph of Proposition 5, as easily checked.

5.3. Exclusive-or $g = h_1 \overline{h}_2 \lor \overline{h}_1 h_2$

Proposition 6. An extension of type $g = h_1 \overline{h}_2 \lor \overline{h}_1 h_2$ exists if and only if every cycle of the structure hypergraph $H$ contains an even number of true-edges (and, since $H$ is bipartite, an even number of false-edges).

Proof. Let us observe now that such an extension exists if and only if there are binary assignments $h_1$ and $h_2$ to the vertices of $V_1$ and $V_2$, respectively, satisfying the following properties:

\[(a, b) \in E_1 \Rightarrow h_1(a) \neq h_2(b),\]
\[(a, b) \in E_0 \Rightarrow h_1(a) = h_2(b).\]

If the graph $H$ contains a cycle let us go around this cycle. Since the edges of $E_0$ keep the parity of the assignment, while the edges of $E_1$ change the parity, according to (2), the parity has to change as many times as the number of true-edges along the cycle. Therefore a contradiction is obtained whenever the cycle contains an odd number of true-edges.

Conversely, if every cycle of $H$ contains an even number of true-edges, then let us assign binary values to all the vertices of $H$ according to the rule (2): Pick a vertex and set its assignment arbitrarily in every connected component of $H$, and propagate this assignment to all other vertices along the edges in each of the components by applying rule (2). Since no cycle with an odd number of true-edges exists, this process does not lead to a contradiction, and hence provides a feasible assignment.

In Example 2 above, the graph $H$ does not have a cycle with an odd number of $E_1$ edges. Hence, picking vertices, say $(1, 1, 1)$ and $(0, 1, 1) \in V_1$ from each of the connected components of the graph $H$, and assigning arbitrary binary values to them, e.g., $h_1(1, 1, 1) = 1$ and $h_1(0, 1, 1) = 0$, the assignment can be completed to a feasible one by following the edges and applying rule (2). It is easy to see that with these definitions of $h_1$ and $h_2$

\[f = h_1 \overline{h}_2 \lor \overline{h}_1 h_2\]

is an extension of $(T, F)$.

Proposition 7. Given a pdBf $(T, F)$, the existence of an extension of scheme $f = g(h_1(S_1), h_2(S_2))$ can be checked in polynomial time.
Proof. The proof proceeds as follows:

1. First we construct the structure graph $H$.
2. Next we check whether $H$ has a subgraph of four vertices $a, b, c, d$ such that $a, c \in V_1, b, d \in V_2, (a, b), (c, d) \in E_1$ and $(c, b) \in E_0$. This can be done by assigning 1 to every vertex incident with a true-edge, and 0 to every other vertex, and checking the false-edges whether each of them is incident with a vertex having a 0 assigned. If yes, i.e. if there is no such subgraph in $H$, then the above assignment is a feasible assignment, and we can STOP (see Proposition 4).
3. If not, then we do the same for the subgraph $a, c \in V_1, b, d \in V_2, (a, b), (c, d) \in E_0$ and $(c, b) \in E_1$. Analogously to the above, this can be done by assigning 0 to every vertex incident with a false-edge, and 1 to every other vertex, and checking the true-edges whether each of them is incident with a vertex having a 1 assigned. If yes, then we have a feasible assignment, and we can STOP (see Proposition 5).
4. If neither of the above yield a feasible assignment, then we check if $H$ has a cycle with an odd number of $E_1$ edges. This can be done by assigning an arbitrary binary value to an arbitrarily selected vertex in every connected component of $H$, and propagate this assignment to all other vertices along the edges in each of the components by applying rules (2). If no contradiction arises, then we obtained a feasible assignment (see Proposition 6). Otherwise, we can conclude that the given pdBf has no extension of the required scheme.

Each of the above steps can obviously be executed in time linear in the number of edges in $H$. □

6. Scheme $f = g(S_0, h_1(S_1), h_2(S_2))$

In this section we show that to decide about the existence of an extension of scheme $f = g(S_0, h_1(S_1), h_2(S_2))$ is, in general, NP-complete. For this, we shall reduce to our problem the so called hypergraph 2-coloring problem, known to be NP-complete (see e.g., [11]).

Hypergraph 2-coloring

Instance: Hypergraph $H^* = (V^*, E^*)$, with $|\Delta| = 3$ for every $\Delta \in E^*$.

Question: Is $H^*$ 2-colorable, i.e. does there exist an assignment $c: V^* \rightarrow \{0, 1\}$ such that for every $\Delta \in E^*$ there exist elements $u, v \in \Delta$ with $c(u) = 1$ and $c(v) = 0$?

In the reduction the small pdBf $(T', F')$ of Fig. 6 will play a key role. The letters $a, b, \ldots, r$ in this pdBf refer to different Boolean vectors of conformable dimensions. It is easy to verify by Proposition 2 that this pdBf $(T', F')$ has no extension of scheme $f' = g'(S_0, h_1(S_1))$. (In the corresponding conflict graph $G$, the vertices corresponding to $p, q$ and $r$ form a triangle; hence $G$ is not bipartite.)

Given a hypergraph $H^* = (V^*, E^*)$, we construct an instance of pdBf $(T, F)$ as follows. To every $\Delta = \{u, v, w\} \in E^*$, we shall associate vertices $a_\Delta, b_\Delta, c_\Delta, p_\Delta, q_\Delta$ and $r_\Delta$, and define true- and false-edges as in Fig. 7.
Fig. 6. A small pdBf which has no extension of scheme \( f = g(S_0, h_1(S_1)) \).

Fig. 7. Construction of true- and false-edges from a triple \( A = \{u, v, w\} \in E^* \).

Formally, we define the structure hypergraph \( H = (V_0 \cup V_1 \cup V_2, E_0 \cup E_1) \) by setting

\[
V_0 = \bigcup_{\Delta \in E^*} \{a, b, c\}, \quad V_1 = \bigcup_{\Delta \in E^*} \{p, q, r\}, \quad V_2 = V^*;
\]

\[
E_1 = \{(a, p, u), (b, q, v), (c, r, w) \mid \Delta = \{u, v, w\} \in E^*\}, \quad (3)
\]

\[
E_0 = \{(a, q, w), (b, r, u), (c, p, v) \mid \Delta = \{u, v, w\} \in E^*\}.
\]

Let finally, \((T, F)\) be a pdBf which has \( H \) as its structure hypergraph. (Obviously, using \(|S_2| > \lceil \log |V_2| \rceil = \lceil \log |V^*| \rceil \) variables, one can construct a binary encoding, which associates different binary vectors of length \(|S_2|\) to the vertices in \( V_2 \). Analogously, using \(|S_0| = |S_1| = \lceil \log |V_0| \rceil = \lceil \log |V_1| \rceil = \lceil \log 3 |E^*| \rceil \) variables, the vertices of \( V_0 \) and \( V_1 \) can also be encoded as different binary strings.)

We claim that the pdBf \((T, F)\) has an extension of scheme \( f = g(S_0, h_1(S_1), h_2(S_2)) \) if and only if \( h_2 \) induces a proper 2-coloring of the hypergraph \( H^* \). For, if \( h_2 \) is not defining a proper 2-coloring of \( H^* \), then there exists a hyperedge \( \Delta = \{u, v, w\} \in E^* \) for
which \( h_2(u) = h_2(v) = h_2(w) \). This implies that the pdBf \((T', F')\), as shown in Fig. 8, must have an extension of the scheme \( f = g'(S_0, h_1(S_1)) \), which is however impossible by Proposition 2, as we observed above.

Conversely, let \( h_2 \) be any binary assignment to the vertices \( V^* = V_2 \), defining a proper 2-coloring of \( H^* \). We show below that this assignment of \( V_2 \) can be extended to \( V_1 \) proving the existence of an extension of the pdBf \((T, F)\). It follows from definitions (3) that for every vertex \( s \in V_1 \) there exists a unique triple \( \Delta \in E^* \) for which \( s \in \{p_\Delta, q_\Delta, r_\Delta\} \), and there exists a unique vertex \( t \in \Delta \), such that \( \{s, t\} \subset e \) for some true-edge \( e \in E_1 \). Let us define \( h_1 \) by \( h_1(s) = h_2(t) \). From the assumption that \( h_2 \) induces a proper 2-coloring of \( V_2 \), it is easy to see that any two hyperedges \( e \in E_2 \), it is easy to see that any two hyperedges \( e \in E_1 \), and \( e' \in E_0 \) of \( H \), with \( e \cap V_0 = e' \cap V_0 = d \), have different pairs of bits assigned by \((h_1, h_2)\). That is, if \( e = (d, s, t) \) and \( e' = (d, s', t') \), then either \( h_1(s) \neq h_1(s') \) or \( h_2(t) \neq h_2(t') \). By Proposition 1, this means that \((T, F)\) has an extension of the desired scheme.

The arguments above prove that the pdBf \((T, F)\) has an extension of scheme \( f = g(S_0, h_1(S_1), h_2(S_2)) \) if and only if the hypergraph \( H^* \) has a proper 2-coloring. Since the decision problem about the existence of an extension of scheme \( f = g(S_0, h_1(S_1), h_2(S_2)) \) is obviously in \( NP \), it follows that:

**Proposition 8.** Deciding if a pdBf \((T, F)\) has an extension of scheme \[ f = g(S_0, h_1(S_1), h_2(S_2)) \]
is \( NP \)-complete.

7. **Scheme \( f = g(h_1(S_1), h_2(S_2), h_3(S_3)) \)**

By an argument similar to the one in the previous section, i.e. by reducing the hypergraph 2-coloring problem to this case, we show that this problem is also \( NP \)-complete. As in the previous construction, the key in this reduction will be again the structure of Fig. 6.

Let us describe the construction of a structure hypergraph \( H \) and the corresponding pdBf \((T, F)\) from a given hypergraph \( H^* = (V^*, E^*) \), in which \(|\Delta| = 3\) holds for every edge \( \Delta \in E^* \).
First, we associate to every edge \( \Delta = \{u, v, w\} \in E^* \) six new vertices \( a_\Delta, b_\Delta, c_\Delta, p_\Delta, q_\Delta, \) and \( r_\Delta, \) and define

\[
V_1 = \bigcup_{\Delta \in E^*} \{a_\Delta, b_\Delta, c_\Delta\}, \quad V_2 = \bigcup_{\Delta \in E^*} \{p_\Delta, q_\Delta, r_\Delta\}, \quad V_3 = V^*.
\] (4)

The sets of true-edges \( E_1 \) and false-edges \( E_0 \) of the structure hypergraph \( H = (V_1 \cup V_2 \cup V_3, E_0 \cup E_1) \) are then given by

\[
E_1 = \{(a_\Delta, p_\Delta, u), (b_\Delta, q_\Delta, v), (c_\Delta, r_\Delta, w) | \Delta = \{u, v, w\} \in E^* \},
\]

\[
E_0 = \{(a_\Delta, q_\Delta, w), (b_\Delta, r_\Delta, u), (c_\Delta, p_\Delta, v) | \Delta = \{u, v, w\} \in E^* \}.
\] (5)

Finally, we let a pdBF \( (T, F) \) have \( H \) as its structure hypergraph. Without any loss of generality we may assume that the sets \( S_1, S_2 \) and \( S_3 \) of scheme \( f = g(h_1(S_1), h_2(S_2), h_3(S_3)) \) are disjoint and that \( |S_1| = |S_2| = \lceil \log 3 |E^*| \rceil \) and \( |S_3| = \lceil \log |V^*| \rceil \), since \( |V_1| = |V_2| = 3 |E^*| \) and \( |V_3| = |V^*| \), and thus all of the vertices can be encoded as distinct binary vectors of appropriate dimensions.

We can show again, as in the previous section, that if all of the vertices of some hyperedge \( \Delta \in E^* \) are assigned the same binary value \( \alpha \), then no extension of scheme \( f = g(\alpha, h_2(S_2), h_3(S_3)) \) exists. This implies that, in any extension of \( (T, F) \) of scheme \( f = g(h_1(S_1), h_2(S_2), h_3(S_3)) \), the function \( h_3 \) must induce a proper 2-coloring of the hypergraph \( H^* \).

On the other hand, if \( h_3 \) defines a 2-coloring for \( H^* \), then, by setting \( h_1(d) = h_2(s) = h_3(t) \) for all \( (d, s, t) \in E_1 \), we can define \( h_1 \) and \( h_2 \) uniquely on \( V_1 \) and \( V_2 \). For illustration, consider the example in Fig. 9.

In view of the definition of \( h_1 \) and \( h_2 \) it is not difficult to see now that for every true-edge \( (d, s, t) \in E_1 \) we have \( (h_1(d), h_2(s), h_3(t)) \in \{0, 0, 0\}, \{1, 1, 1\} \), but for every false-edge \( (d', s', t') \in E_0 \) the vector \( (h_1(d'), h_2(s'), h_3(t')) \) does not belong to \( \{0, 0, 0\}, \{1, 1, 1\} \), i.e., no binary triple encodes simultaneously a true-edge and a false-edge of \( H \). By Proposition 1, this implies the existence of an extension of scheme \( f = g(h_1(S_1), h_2(S_2), h_3(S_3)) \).

Consequently, the pdBF \( (T, F) \) has an extension of scheme \( f = g(h_1(S_1), h_2(S_2), h_3(S_3)) \) if and only if the hypergraph \( H^* \) is 2-colorable. Since the decision problem about the existence of an extension of this scheme is obviously in NP, we have:

**Proposition 9.** Deciding if a pdBF \( (T, F) \) has an extension of scheme

\[
f = g(h_1(S_1), h_2(S_2), h_3(S_3))
\]

is NP-complete.
8. Tractable cases of scheme \( f = g(S_0, h_1(S_1), h_2(S_2)) \)

In this section we present an algorithm that, given a pdBf \((T,F)\), decides the existence of an extension of scheme \( f = g(S_0, h_1(S_1), h_2(S_2)) \), even though the problem was shown to be NP-complete, in general. We also show that if

\[
|S_0| = O(\log \log m + \log \log n),
\]

where \( m = |T \cup F| \) and \( n = |V| \), then this algorithm runs in polynomial time.

Let us introduce the notations \( T(g) \) and \( F(g) \) for the sets of all true and all false vectors of a Boolean function \( g \), respectively. To illustrate the basic idea of the algorithm, consider first a simple observation on Boolean functions in two variables:

**Proposition 10.** For any Boolean function \( g \) in two variables, the sets \( T(g) \) and \( F(g) \) can be characterized by quadratic Boolean equations. The number of equations for each set is not more than 3.

For example, if \( g = \bar{x}y \lor xy \) (exclusive-or), then the sets

\[
T(g) = \{(0,1),(1,0)\} \quad \text{and} \quad F(g) = \{(0,0),(1,1)\}
\]

can be characterized by quadratic Boolean equations, e.g. \( T(g) \) by \( \bar{x}y = xy - 0 \), and \( F(g) \) by \( \bar{x}y = x\bar{y} = 0 \). Similar arguments can be applied to any Boolean function in two variables.
Given a pdBf \((T, F)\) and the scheme \(f = g(S_0, h_1(S_1), h_2(S_2))\), we consider its structure hypergraph \(H = (V, E)\). By Proposition 1, an extension of scheme \(f = g(S_0, h_1(S_1), h_2(S_2))\) exists if and only if there are binary assignments \(h_1\) and \(h_2\) to the vertices of \(V_1\) and \(V_2\) of the structure hypergraph \(H\), respectively, for which \(T_w \cap F_w = \emptyset\) for every \(w \in V_0\), where

\[ T_w = \{(h_1(u), h_2(v)) | (w, u, v) \in E_1\}, \]

\[ F_w = \{(h_1(u), h_2(v)) | (w, u, v) \in E_0\}. \]

In other words, for every \(w \in V_0\) there is an extension (i.e. a Boolean function in two variables) \(g_w(x, y)\) of pdBf \((T_w, F_w)\). Therefore, if we introduce the variables \(X_u = h_1(u)\) for \(u \in V_1\) and \(Y_v = h_2(v)\) for \(v \in V_2\), and if we fix the function \(g_u\), then the sets \(T_w \subseteq T(g_u)\) and \(F_w \subseteq F(g_u)\) can be characterized by the above quadratic equations. For example, if \(g_w = x \lor y\), we generate \(X_w Y_w = 0\), \(X_u Y_v = 0\) for each \((w, u, v) \in E_1\) and \(X_u Y_v = 0\), \(X_u Y_w = 0\) for each \((w, u, v) \in E_0\). We then generate these quadratic equations for all \(w \in V_0\).

Although we do not know the functions \(g_u\) in advance, we can test all possibilities, since there are only \(2^{2^2} = 16\) different functions in two variables.

The following algorithm can be used to conclude whether the pdBf \((T, F)\) has an extension in a desired scheme.

Algorithm G-3

1. Construct the structure hypergraph \(H = (V_0 \cup V_1 \cup V_2, E_0 \cup E_1)\) for the given pdBf \((T, F)\), and associate Boolean variables \(X_u\) and \(Y_v\) to all vertices \(u \in V_1\) and \(v \in V_2\).
2. For each \(w \in V_0\), select a two-variable Boolean function \(g_w\) (there are 16 possibilities for each \(w\), and thus a total number of \(16^{|V_0|}\) cases).
3. For each selection of functions \(g_w, w \in V_0\), set up the following system of quadratic Boolean equations:
   - For every true-edge \((w, u, v) \in E_1\) add the equations characterizing \(T(g_w)\) using the variables \(X_u\) and \(Y_v\).
   - For every false-edge \((w, u, v) \in E_0\) add the equations characterizing \(F(g_w)\) using the variables \(X_u\) and \(Y_v\).

Let \(B = 0\) denote the resulting system of quadratic Boolean equations.
4. If \(B = 0\) has a solution for one of the selections, it shows the existence of an extension of the required scheme. Let in this case \(h_1(u) = X_u\) for \(u \in V_1\) and \(h_2(v) = Y_v\) for \(v \in V_2\), and STOP.
   If none of the selections yields a consistent equation, then we can conclude that the pdBf \((T, F)\) has no extension of the desired scheme.
Since the number of equations in each of the systems is not more than \( 3|E| = 3m \), and since a quadratic Boolean equation can be solved in time linear in its size (see [2]), the running time of algorithm G-3 is \( O(m|V_0|) \). This proves the following result.

**Proposition 11.** Given a pdBf \((T, F)\), the existence of an extension of scheme \( f = g(S_0, h_1(S_1), h_2(S_2)) \) can be decided in polynomial time, if \(|V_0| = O(\max(\log n, \log m))\), or, in particular, if \(|S_0| = O(\max(\log \log n, \log \log m))\).

### 9. Positive schemes

Let us recall that a scheme \( f = g(S_0, h_1(S_1), \ldots, h_k(S_k)) \) is called positive if the functions \( g, h_1, \ldots, h_k \) are all required to be positive. We can associate the structure hypergraph \( H = (V_0 \cup V_1 \cup \cdots \cup V_k, E_0 \cup E_1) \) to a positive scheme in the same manner as to general schemes. However we need to add the following directed arcs:

\[
A_i = \{(u, u') | u, u' \in V_i \text{ for which } u < u'\}, \quad i = 0, 1, \ldots, k.
\]

The resulting hypergraph \( H^+ = (V_0 \cup V_1 \cup \cdots \cup V_k, E_0 \cup E_1, A_0 \cup A_1 \cup \cdots \cup A_k) \) is called the augmented structure hypergraph. In illustrations, the directed arcs of \( H^+ \) will be indicated by arrows.

In order to handle positive schemes, we modify Proposition 1, using the well-known fact that a pdBf \((T, F)\) has a positive extension of scheme \( f = g(S_0) \) if and only if no pair of vectors \( x \in T \) and \( y \in F \) satisfy the inequality \( x[S_0] \leq y[S_0] \).

**Proposition 12.** A pdBf \((T, F)\) has an extension of positive scheme \( f = g(S_0, h_1(S_1), \ldots, h_k(S_k)) \) if and only if there are positive binary assignments \( h_i: V_i \to \{0, 1\}, \quad i = 1, \ldots, k \) (i.e. assignments for which \( h_i(u) \leq h_i(v) \) holds whenever \( u \leq v \) for some \( u, v \in V_i \)), such that no pair of true- and false-edges \((u_0, u_1, \ldots, u_k) \in E_1 \) and \((v_0, v_1, \ldots, v_k) \in E_0 \) satisfies \((u_0, h_1(u_1), \ldots, h_k(u_k)) \leq (v_0, h_1(v_1), \ldots, h_k(v_k))\).

### 10. Positive scheme \( f = g(S_0, h_1(S_1)) \)

Given a pdBf \((T, F)\), let us denote by \( T^* \) and \( F^* \) the following subsets:

\[
T^* = \{x \in T | \exists y \in F \text{ satisfying } y[S_0] \geq x[S_0]\}, \\
F^* = \{y \in F | \exists x \in T \text{ satisfying } y[S_0] \geq x[S_0]\}.
\]

**Proposition 13.** A pdBf \((T, F)\) has an extension of positive scheme \( f = g(S_0, h_1(S_1)) \) if and only if there is no pair of vectors \( x \in T^* \) and \( y \in F^* \), such that \( x[S_1] \leq y[S_1] \).
Proof. Assume that there is a pair of vectors, \( x \in T^* \) and \( y \in F^* \), for which there are \( x' \in T \) and \( y' \in F \) such that \( x[S_0] \leq y[S_1], x[S_0] \leq y'[S_0] \) and \( x'[S_0] \leq y[S_0] \) (see the corresponding subgraph of the augmented structure hypergraph \( H^+ \) in Fig. 10). Then \( h(x[S_1]) = 1 \) must hold, since otherwise, the inequality \( (y'[S_0], h_1(y'[S_1])) \geq (x[S_0], h_1(x[S_1])) \) would follow, disproving the existence of a positive extension by Proposition 12. Then, the positivity of \( h_1 \) implies \( h_1(y[S_1]) = 1 \). Thus, the inequality \( (y[S_0], h_1(y[S_1])) \geq (x'[S_0], h_1(x'[S_1])) \) follows, contradicting the existence of a positive extension by Proposition 12. Therefore, if \((T,F)\) has the desired extension, there cannot exist such a pair of vectors, thus, proving the first half of the statement.

On the other hand, if there is no such pair of vectors, then let us define \( h_1 \) by

\[
h_1(v) = \begin{cases} 
1, & \text{if } v \in V_1 \text{ and } \exists x \in T^* \text{ for which } x[S_1] \leq v, \\
0, & \text{otherwise.} 
\end{cases}
\]

One can verify that the resulting \( h_1 \) satisfies the conditions of Proposition 12.

Example 3. Let us consider the pdBf of Fig. 11. For this pdBf, we obtain \( T^* = \{(0,1,1,0,1),(0,1,1,1,0)\} \) and \( F^* = \{(0,1,0,1,0)\} \), and the conditions of Proposition 13 are satisfied. Thus, the desired extension exists. The last column in Fig. 11 lists the values of \( h_1 \).

![Fig. 10. The augmented structure hypergraph \( H^+ \) used in the proof of Proposition 13.](image)

<table>
<thead>
<tr>
<th></th>
<th>( S_0 )</th>
<th>( S_1 )</th>
<th>( h_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>1 1 0 1 1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0 1 1 0 1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 1 1 1 0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( F )</td>
<td>0 0 1 0 1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 0 1 1 0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

![Fig. 11. The pdBf \((T,F)\) of Example 3.](image)
Since the existence of the subgraph of Fig. 10 in the augmented structure hypergraph $H^+$ can be tested in polynomial time, we conclude that:

**Proposition 14.** Deciding if a pdBf $(T, F)$ has an extension of positive scheme $f = g(S_0, h_1(S_1))$ can be done in polynomial time.

11. Positive scheme $f = g(h_1(S_1), h_2(S_2))$

**Proposition 15.** A pdBf $(T, F)$ has an extension of positive scheme $f = g(h_1(S_1), h_2(S_2))$ if and only if its augmented structure hypergraph $H^+$ does not contain simultaneously the subgraphs shown in Fig. 12.

**Note.** Some or all of the vertices within $V_1$ (or within $V_2$) of the subgraphs in Fig. 12 may coincide.

**Proof of Proposition 15.** Let us observe that the existence of the subgraph shown in Fig. 12(a) in $H^+$ implies that $h_1(a_1[S_1]) = 1$, because otherwise $(h_1(a_1[S_1]), h_2(a_1[S_2])) \leq h_1(a_3[S_1]), h_2(a_3[S_2]))$ follows by the relation $a_1[S_2] \leq a_3[S_2]$ and by the positivity of $h_2$. Since $a_1 \in T$ and $a_3 \in F$, this contradicts the positivity of $g$. Analogously, $h_2(a_1[S_2]) = 1$ is implied by $a_1[S_1] \leq a_2[S_1]$, and by the positivity of $h_1$ and $g$. Since the assignments $h_1$ and $h_2$ must distinguish the true vector $a_1 \in T$ from the false vectors $a_2, a_3 \in F$, and since $h_1(a_2[S_1]) = h_2(a_3[S_2]) = 1$ are already implied, $h_1(a_3[S_1]) = h_2(a_2[S_2]) = 0$ must hold. In other words, the existence of the subgraph shown in Fig. 12(a) implies the following relations:

\[
\{(h_1(a[S_1]), h_2(a[S_2])) | a \in T\} = \{(1, 1)\},
\]

\[
\{(h_1(a[S_1]), h_2(a[S_2])) | a \in F\} \supseteq \{(1, 0), (0, 1)\}.
\]

**Fig. 12.** The forbidden subgraphs of Proposition 15.
Note that \((h_1(a[S_1]), h_2(a[S_2])) \neq (0, 0)\) for all \(a \in T\) follows from the positivity of \(g\), as long as \(F \neq \emptyset\).

Similarly to the above, if \(H^+\) contains the subgraph 12(b), then we have
\[
\{(h_1(a[S_1]), h_2(a[S_2]))| a \in F\} = \{(0, 0)\},
\]
\[
\{(h_1(a[S_1]), h_2(a[S_2]))| a \in T\} \supseteq \{(1, 0), (0, 1)\}.
\]

However, conditions (7) and (8) are not consistent, and thus a pdBf for which \(H^+\) contains both subgraphs cannot have an extension of the desired positive scheme.

To show the converse direction of the statement, let us assume for instance that \(H^+\) does not contain the subgraph 12(b). Let us then define \(h_1(v)\) for \(v \in V_1\) and \(h_2(w)\) for \(w \in V_2\), as follows:
\[
h_1(v) = \begin{cases} 
1, & \text{if } a[S_1] \leq v \text{ for some } a \in T, \\
0, & \text{otherwise},
\end{cases}
\]
\[
h_2(w) = \begin{cases} 
1, & \text{if } a[S_2] \leq w \text{ for some } a \in T, \\
0, & \text{otherwise}.
\end{cases}
\]

With this definition \((h_1(a[S_1]), h_2(a[S_2])) = (1, 1)\) for all true vectors \(a \in T\), and no false vector has the same assignment, since otherwise the subgraph 12(b) could be found in \(H^+\).

If the subgraph shown in 12(a) is missing from \(H^+\), then the analogous assignment
\[
h_1(v) = \begin{cases} 
0, & \text{if } b[S_1] \geq v \text{ for some } b \in F, \\
1, & \text{otherwise},
\end{cases}
\]
\[
h_2(w) = \begin{cases} 
0, & \text{if } b[S_2] \geq w \text{ for some } b \in F, \\
1, & \text{otherwise},
\end{cases}
\]
for \(v \in V_1\) and \(w \in V_2\), will work. □

**Example 4.** Let us consider the pdBf \((T, F)\) of Fig. 13, whose augmented structure hypergraph is shown in Fig. 14.

The graph \(H^+\) of Fig. 14 does not contain the subgraph 12(b), although it does contain the subgraph 12(a) (the vertices (101) and (010) of \(V_1\), and (01) and (00) of \(V_2\) form a degenerate version of the subgraph 12(a)). Therefore, we shall use the assignment (9). This yields the last three columns of Fig. 13. It is easy to see that these functions \(h_1\) and \(h_2\) are both positive, and that \(g = h_1 \land h_2\).

Since checking the existence of the subgraphs of the previous proposition is obviously a polynomial procedure, we can conclude:

**Proposition 16.** Deciding if a pdBf \((T, F)\) has an extension of positive scheme \(f = g(h_1(S_1), h_2(S_2))\) can be done in polynomial time.
12. Positive scheme $f = g(S_0, h_1(S_1), h_2(S_2))$

Let us consider the augmented structure hypergraph of a given pdBf $(T, F)$, and associate Boolean variables $X_u = h_1(u)$ and $Y_v = h_2(v)$ to $u \in V_1$ and $v \in V_2$. The positivity of $h_1$ and $h_2$ implies that the following equations must hold:

$$X_u X_{u'} = 0 \quad \forall u, u' \in V_1 \text{ such that } u \leq u',$$

$$Y_v Y_{v'} = 0 \quad \forall v, v' \in V_2 \text{ such that } v \leq v'. \quad (11)$$

Let us then observe that $X_u = Y_v = 0$ is possible for a true edge $(w, u, u') \in E_1$ only if no false edge is incident with any of the vertices $w' \in V_0$ for which $w' \geq w$; otherwise $g$ could not be positive. On the other hand, if $w \in V_0$ satisfies this condition, these true edges $(w, u, v) \in E_1$ are not confused with any false edge, since their $V_0$ segments distinguish them properly. Thus, such vertices $w \in V_0$ do not impose any restriction
on the solution. Therefore, \((T,F)\) has an extension of the required positive scheme if and only if \((T',F)\) has one, \(T'\) being obtained from \(T\) by the deletion of all the true vectors \(x \in T\) for which no false vector \(y \in F\) satisfies \(y[S_0] \geq x[S_0]\).

Analogously, \(X_u = Y_v = 1\) is possible for a false edge \((w,u,v) \in E_0\) only if no true edge is incident with any of the \(w' \in V_0\) vertices for which \(w \geq w'\). Similarly to the above argument we can conclude that \((T',F')\) has an extension of this scheme if and only if \((T,F')\) has one, \(F'\) being obtained from \(F\) by the deletion of all the false vectors \(y \in F\) for which no true vector \(x \in T'\) satisfies \(y[S_0] > x[S_0]\).

Obviously, these preprocessing steps can be performed in polynomial time.

Based on the above reductions, we can now assume that for any true edge \((w,u,v) \in E_1\) there exists a false edge \((w',u',u') \in E_0\) with \(w' \geq w\), and conversely, for any false edge \((w',u',u') \in E_0\) there exists a true edge \((w,u,v) \in E_1\) with \(w \leq w'\). This implies that the following equations must hold:

\[
X_u Y_v = 0 \quad \text{whenever there is a } w \in V_0 \text{ for which } (w,u,v) \in E_1, \tag{12}
\]

\[
X_u Y_v = 0 \quad \text{whenever there is a } w \in V_0 \text{ for which } (w,u,v) \in E_0.
\]

Let us introduce four Boolean variables for every vertex \(w \in V_0\), denoted by \(T_{w,\alpha}\) and \(F_{w,\alpha}\) for \(\alpha \in \{0,1\}\), with the meaning that \(T_{w,\alpha} = 1\) holds exactly when there is a true edge \((w,u,v) \in E_1\) for which \(X_u = \alpha\) and \(Y_v = \bar{\alpha}\), and \(F_{w,\alpha} = 1\) holds exactly when \(X_u = \alpha\) and \(Y_v = \bar{\alpha}\) for some false edge \((w,u,v) \in E_0\). The positivity of \(g\) implies that \(T_{w,\alpha} F_{w',\alpha} = 1\) is not feasible for any comparable pair of vertices \(w, w' \in V_0\) for which \(w \leq w'\). Thus the following equations must also hold for any solution:

\[
T_{w,\alpha} F_{w',\alpha} = 0 \quad \text{for all } w, w' \in V_0 \text{ with } w \leq w' \text{ and for all } \alpha \in \{0,1\}. \tag{13}
\]

Let us observe finally that if \((w,u,v) \in E_1\), then \(X_u = 0\) implies \(Y_v = 1\) by (12), and thus \(T_{w,0} = 1\) follows. Similarly, \(Y_v = 0\) implies \(T_{w,1} = 1\). Analogously, if \((w,u,v) \in E_0\), then \(X_u = 1\) implies \(F_{w,1} = 1\), and \(Y_v = 1\) forces \(F_{w,0} = 1\). Summarizing these relations, the following equations must hold:

\[
T_{w,0} X_u = 0 \quad \text{and} \quad T_{w,1} Y_v = 0 \quad \text{for } (w,u,v) \in E_1, \tag{14}
\]

\[
F_{w,1} X_u = 0 \quad \text{and} \quad F_{w,0} Y_v = 0 \quad \text{for } (w,u,v) \in E_0.
\]

We shall show next that the above system of quadratic Boolean equations (11)-(14) is not only necessary but also sufficient for the existence of the desired positive scheme.

**Proposition 17.** A pdBf \((T,F)\) has an extension of positive scheme \(f = g(S_0, h_1(S_1), h_2(S_2))\) if and only if the system of Boolean equations (11)-(14) has a solution.

**Proof.** We have shown above that all the existing extensions of the desired positive scheme correspond to feasible solutions of equations (11)-(14). It is enough now to prove that if (11)-(14) has a solution, then by defining \(h_1(u) = X_u\) and \(h_2(v) = Y_v\) for all \(u \in V_1\) and \(v \in V_2\), the list of true vectors \((w,h_1(u),h_2(v))\) for \((w,u,v) \in E_1\) and false vectors \((w',h_1(u'),h_2(v'))\) for \((w',u',v') \in E_0\) has a positive extension \(g\). In other words,
we have to show that for every pair of true edges \((w, u, v) \in E_1\) and false edges \((w', u', v') \in E_0\) the relations \(w \leq w', h_1(u) \leq h_1(u')\), and \(h_2(v) \leq h_2(v')\) cannot hold simultaneously.

Let us assume indirectly that there is a pair of true and false edges, \((w, u, v) \in E_1\) and \((w', u', v') \in E_0\) for which the inequalities

\[
\begin{align*}
w &< w', \\
x_u &< x_{u'}, \\
y_v &< y_{v'}.
\end{align*}
\]

hold. Since (12) implies that \(X_u + Y_v \geq 1\) for every true edge \((w, u, v) \in E_1\) and that \(X_{u'} + Y_{v'} \leq 1\) for every false edge \((w', u', v') \in E_0\), it follows from (15) that \(X_u = X_{u'} = \alpha\) and \(Y_v = Y_{v'} = \beta\) for some \(\alpha \in \{0, 1\}\). Then conditions (14) imply that \(T_{w, v} = F_{w', v'} = 1\), which together with the relation \(w' \geq w\) lead to a contradiction with (13).

In contrast to the NP-completeness result of Proposition 8, we get now:

**Proposition 18.** Given a pdBf \((T, F)\), the existence of an extension of positive scheme \(f = g(S_0, h_1(S_1), h_2(S_2))\) can be decided in polynomial time.

**Proof.** The Boolean equations (11)–(14) are quadratic and contain a polynomial number of variables and equations. Since a quadratic Boolean equation is solvable in linear time (see [2]), the consistency of the above system is polynomially decidable.

13. **Positive scheme** \(f = g(h_1(S_1), h_2(S_2), h_3(S_3))\)

Let us consider the structure hypergraph \(H = (V_1 \cup V_2 \cup V_3, E_0 \cup E_1)\) for a given pdBf \((T, F)\), and let us introduce the Boolean variables \(X_u, Y_v, Z_w\) defined as:

\[
\begin{align*}
X_u &= h_1(u), \quad u \in V_1, \\
Y_v &= h_2(v), \quad v \in V_2, \\
Z_w &= h_3(w), \quad w \in V_3.
\end{align*}
\]

We shall distinguish the following three cases:

(i) \(\exists (u, v, w) \in E_0: X_u + Y_v + Z_w \geq 2\),

(ii) \(\exists (u', v', w') \in E_1: X_{u'} + Y_{v'} + Z_{w'} \leq 1\),

(iii) \(\forall (u, v, w) \in E_0: X_u + Y_v + Z_w \leq 1\) and \(\forall (u', v', w') \in E_1: X_{u'} + Y_{v'} + Z_{w'} \geq 2\).

**Case (i):** Let us assume that the inequality \(X_u + Y_v + Z_w \geq 2\) holds for a false edge \((u, v, w) \in E_0\). Without loss of generality we consider the case of \(X_u = Y_v = 1\). If \(Z_w = 1\), then \(T = \emptyset\) is implied, otherwise we get a contradiction with the positivity of \(g\).
Thus, assuming \( F \neq \emptyset \) and \( T \neq \emptyset \), we must have \( Z_w = 0 \). Furthermore, by the positivity of \( g \) and \( h_3 \), we have the following observations:

- There cannot exist a true edge \( (u', v', w') \in E_1 \) for which the inequality \( w > w' \) holds.
- Every vertex \( w' \in V_3 \), for which there exists a true edge \( (u'', v'', w'') \in E_1 \) with \( w' > w'' \), we must have \( Z_{w'} = 1 \).
- To every vertex \( w' \in V_3 \) for which there is no true edge \( (u'', v'', w'') \in E_1 \) with \( w' > w'' \), we are free to assign \( Z_{w'} = 0 \) (i.e. if there is a solution for the problem, then by changing to \( h_3(w') = 0 \) for all these vertices \( w' \in V_3 \), we obtain again a solution).

We can notice that, by the above observations, every vertex \( w' \in V_3 \) will be assigned a unique binary value \( Z_{w'} \), unless there exists a true edge \( (u', v', w') \in E_1 \) with \( w' < w \). In the latter case we can conclude that there is no solution with the above assumption.

Then, since all vertices \( w \in V_3 \) have now binary values assigned, the structure hypergraph \( H^+ \) can be simplified by merging the vertices in \( V_3 \) with the same binary value into one, or equivalently, by aggregating the columns in \( S_3 \) into one. In this way we could reduce our problem to the problem of finding an extension of positive scheme \( f = g(S_0, h_1(S_1), h_2(S_2)) \), where \( S_0 \) consists now only of one variable, and such a problem is polynomially decidable by Proposition 18.

Repeating the above for all possible selections of the false edge \( (u, v, w) \in E_0 \), and analogously for the remaining cases \( X_w = Z_w = 1 \) and \( Y_v = Z_v = 1 \), we can decide the existence of a solution of type (i) in polynomial time.

Case (ii): This case can be handled in a dual way to case (i), also in polynomial time.

Case (iii): It is easy to observe that all solutions of this type must satisfy the following equations:

\[
X_u X_{u'} = 0 \quad \text{for all } u, u' \in V_1 \text{ for which } u \leq u',
\]

\[
Y_u Y_{v'} = 0 \quad \text{for all } u, v' \in V_2 \text{ for which } v \leq v',
\]

\[
Z_w Z_{w'} = 0 \quad \text{for all } w, w' \in V_3 \text{ for which } w \leq w',
\]

\[
X_u Y_v = 0, \quad Y_v Z_w = 0, \quad Z_w X_u = 0, \quad \text{for all } (u, v, w) \in E_0,
\]

\[
X_{u'} Y_{v'} = 0, \quad Y_{v'} Z_{w'} = 0, \quad Z_{w'} X_{u'} = 0, \quad \text{for all } (u', v', w') \in E_1.
\]

It is also immediate to observe that any solution of (16) and (17) defines indeed a solution.

Since the system of Boolean equations (16) and (17) is a quadratic system, and since 2-SAT is solvable in linear time [2], we can conclude that the existence of a solution of this case can also be decided in polynomial time.

In contrast to the NP-completeness result of Proposition 9, we have now

**Proposition 19.** Deciding if a pdBf \((T, F)\) has an extension of positive scheme \( f = g(h_1(S_1), h_2(S_2), h_3(S_3)) \) can be done in polynomial time.
Table 1
Summary of results

<table>
<thead>
<tr>
<th>Scheme</th>
<th>General extensions</th>
<th>Monotone extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f = g(S_0, h_1(S_1))$</td>
<td>$O(mn)$</td>
<td>$O(m^2n)$</td>
</tr>
<tr>
<td>$f = g(h_1(S_1), h_2(S_2))$</td>
<td>$O(mn)$</td>
<td>$O(m^2n)$</td>
</tr>
<tr>
<td>$f = g(S_0, h_1(S_1), h_2(S_2))$</td>
<td>$O(m^{16^{1^{\cdot^{1}}}})$</td>
<td>$O(m^2n)$</td>
</tr>
<tr>
<td>$f = g(h_1(S_1), h_2(S_2), h_3(S_3))$</td>
<td>NP-complete</td>
<td>$O(m^3n)$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$f = g(S_0, h_1(S_1), \ldots, h_k(S_k))$ ($k \geq 3$)</td>
<td>NP-complete</td>
<td>?</td>
</tr>
<tr>
<td>$f = g(h_1(S_1), \ldots, h_k(S_k))$ ($k \geq 4$)</td>
<td>NP-complete</td>
<td>?</td>
</tr>
</tbody>
</table>

14. Conclusions

In this paper we studied the decomposability of partially defined Boolean functions. In Table 1 we summarize the complexity of the recognition problems of the various schemes we considered.

There are many open problems left for future research, such as the complexity of recognizing general scheme decomposability of monotone extensions, or considering the decomposability of other types of practically important extensions, e.g. Horn extensions, quadratic extensions, etc.

An even more important algorithmic problem is finding efficiently the subsets $S_0, S_1, \ldots, S_k$ of a scheme for which the given partially defined Boolean function has a decomposable extension. For completely specified monotone Boolean functions there are efficient algorithms for finding such decompositions (or concluding that the given function has none), see e.g. [18]. It would be interesting to generalize those ideas for the case of partially defined Boolean functions.

Let us conclude finally with a positive remark about the decomposability of partially defined Boolean functions. A decomposability related complexity measure was introduced in [1], and it was shown that completely specified Boolean functions with a "small" number of zeros tend to have nontrivial decompositions. Although this result is existential and not algorithmic, it suggests that partially defined Boolean functions tend to have nontrivially decomposable extensions, since the set of specified false points usually has a much smaller cardinality than $2^n$.

Acknowledgements

The authors are thankful to Ondřej Čepek for the many helpful comments made on an earlier version of this paper.
References