COMPATIBILITY MATRIX METHOD

SERGEY GUBIN

ABSTRACT. Article presents the compatibility matrix method and illustrates it with the application to P vs NP problem. The method is a generalization of descriptive geometry: in the method, we draft problems and solve them utilizing the image creation technique. The method reveals: P = NP = PSPACE ⊊ P/poly, etc.

INTRODUCTION

This article presents the *compatibility matrix method* [1, 2, 3, 4] and some of the results reported by the author at The Midwest Conference on Combinatorics, Cryptography and Computing (MCCCC) [5, 6, 7].

The compatibility matrix method is a generalization of descriptive geometry on the combinatorial problems: the problems are "drafted" at first and then solved with the image creation technique. We prove: the blueprints' size is polynomial in the problems' NTIME, and the image creation technique's computational complexity is polynomial in the blueprints' size. And we apply these results to the P vs NP problem [8].

1. COMPATIBILITY MATRIX

The following definition formalizes the notion of problem's blueprints.

Definition 1.1. *Compatibility matrix B* is a symmetric Boolean box matrix with the diagonal Boolean matrices on their major diagonal:

$$(1.1) \qquad \begin{cases} 1). & B & = & (B_{ij})_{n\times n} \\ 2). & B_{ij} & = & (b^{ij}_{\mu\nu})_{m_i\times m_j} \\ 3). & b^{ij}_{\mu\nu} & \in & \{false, true\} \\ 4). & b^{ij}_{\mu\nu} & = & b^{ji}_{\nu\mu} \\ 5). & b^{ij}_{\mu\nu} & \Rightarrow & i\neq j \ \vee \ \mu=\nu \end{cases}$$

- where numbers n, m_1, m_2, …, m_n, and maximum

$$(1.2) \qquad m = \max_{i\in[n]} m_i$$

are the appropriate sizes of the compatibility matrix. Boxes B_{ij} are *compatibility boxes.* Those components $b^{ij}_{\mu\nu}$ which equal *true* are *true*-components, and the rest components are *false*-components.

Index function of compatibility matrix B is any one-meaning function[1]

$$\phi : i \in [n] \ \mapsto \ \phi(i) \in [m_i]$$

Each index function ϕ defines a 2-dimensional sub-array of the compatibility matrix which is a *grid of components* or just a *grid*:

$$\{b^{ij}_{\phi(i)\phi(j)}\}_{i,j\in[n]}$$

This grid is a *solution grid* if all its components are *true*-components, i.e. if the grid's index function ϕ satisfies the following functional equation:

$$b^{ij}_{\phi(i)\phi(j)} = true, \ i, j \in [n] \tag{1.3}$$

- solution grid in compatibility matrix B is an orthogonal lattice of *true*-components, one component per compatibility box.

Further, a *true*-component $b^{i_0 j_0}_{\mu_0 \nu_0} = true$ is a *noise* if functional equation 1.3 is inconsistent subject to the following constrains:

$$\phi(i_0) = \mu_0, \ \phi(j_0) = \nu_0 \tag{1.4}$$

Otherwise, *true*-component $b^{i_0 j_0}_{\mu_0 \nu_0} = true$ belongs to a solution grid.

Furthermore, inversion of all noisy components in compatibility matrix 1.1 will transform it into its *general solution*[2].

Exercise 1.2. The following Boolean box matrix (as usual, values *true* and *false* are shown with 1 and 0, appropriately) is a compatibility matrix:

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \left(\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 0 & 1 \\ \hline 1 & 1 & 1 \end{array}\right)$$

In this compatibility matrix, $n = 2$, $m = m_1 = 2$, and $m_2 = 1$. So, this compatibility matrix has two index functions (ϕ and ψ) and two grids of components, appropriately:

$$\begin{array}{c||c} \phi(1) = 1, \ \phi(2) = 1 & \psi(1) = 2, \ \psi(2) = 1 \\ \hline\hline \left(\begin{array}{cc|c} 1 & \cdot & 1 \\ \cdot & \cdot & \cdot \\ \hline 1 & \cdot & 1 \end{array}\right) & \left(\begin{array}{cc|c} \cdot & \cdot & \cdot \\ \cdot & 0 & 1 \\ \hline \cdot & 1 & 1 \end{array}\right) \end{array}$$

[1] For any integer number $N \geq 1$, $[N] = \{1, 2, \ldots, N\}$.

[2] We could present any solution grid from compatibility matrix 1.1 as a compatibility matrix of the same box structure as matrix B and whose components all equal *false* except those which belong to the grid. Then, the general solution would be the per-component disjunction of all those "particular solutions": $(a_{ij}) \vee (b_{ij}) = (a_{ij} \vee b_{ij})$.

Index function ϕ creates a solution grid. Components b^{12}_{21} and b^{21}_{12} are noise, and inversion of them transforms this matrix into its general solution:

$$G = \left(\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 1 \end{array}\right) = \left(\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 1 \end{array}\right) \vee \left(\begin{array}{cc|c} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right)$$

- it may be seen as the disjunction of all solution grids of matrix B.

Exercise 1.3. The following Boolean box matrix is another compatibility matrix:

$$B = \left(\begin{array}{cc|cc|cc} 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{array}\right)$$

In this compatibility matrix, $n = 3$ and $m = m_1 = m_2 = m_3 = 2$. Then, there are 8 grids of components in this matrix. Yet, none of them is a solution grid. Then, this matrix's general solution equals $((0)_{2\times2})_{3\times3}$.

Compatibility matrices are used to encode the natural problems and sometimes the whole families of the problems. There are not any requirements to the *compatibility-matrix encoding* except the encoding's *adequacy*:

Protocol 1.4. *There are solution grids in the given compatibility matrix iff the encoded problem is consistent, i.e. iff it has solutions.*

Several examples of the adequate compatibility-matrix encoding are presented in Section 6. Also, Section 6 specifies domain of the encoding.

Due to Protocol 1.4, solution-grids in the compatibility-matrix encoding may be identified with the solutions of the encoded problem in the "topological sense." Then, the compatibility boxes are the solution's 2D-views; the compatibility matrix itself is an orthographic projection layout display; and functional equation 1.3 is the image creation technique.

The compatibility-matrix encoding of the decision and search problems is appropriately the Post and Turing reduction of the problems to the solution grids existence problem (to find a solution grid we may: iterate the compatibility boxes and the *true*-components in the current box; nil all components in the current box and its transpose box except the current component and its transpose component, and test the resulting matrix on the solution grids existence; we move to the next component if the current component is noise, else we move to the next compatibility box).

General solution of the given compatibility matrix is a compatibility matrix of the encoded problem on its own. And, due to its definition, it contains *true*-components iff the given compatibility matrix contains solution grids. General solution is the useful signal carried by the compatibility matrix. To compute the general solution, we need inverse the noisy components. Such a revitalization of the compatibility matrix is called *depletion.*

2. Solution grids and general solution

Let's solve functional equation 1.3 for the given compatibility matrix 1.1 with the compatibility matrix method itself. For that, let's device such an adequate compatibility-matrix encoding of the equation's solutions which will allow its solution with the naive image creation technique. The encoding's idea is illustrated in the following exercise.

Exercise 2.1. [The major observation] Let *true*-component $b^{i_0j_0}_{\mu_0\nu_0} = true$ be a noise in compatibility matrix 1.1. And let ϕ be a partial index function:

$$\phi : i \in s \subset [n] \mapsto \phi(i) \in [m_i]$$

Let $i_0, j_0 \in s$, and let ϕ satisfy "boundary conditions" 1.4. Let's use random walk method and extrapolate partial function ϕ as far as possible.

We iterate set $[n] - \mathrm{Dom}(\phi)$. For the current $i_1 \in [n] - \mathrm{Dom}(\phi)$, we search set $[m_{i_1}]$ for any such $\mu_{i_1} \in [m_{i_1}]$ that

$$b^{i_1i_1}_{\mu_{i_1}\mu_{i_1}} \wedge \bigwedge_{i\in\mathrm{Dom}(\phi)} b^{ii_1}_{\phi(i)\mu_{i_1}} = true$$

When such μ_{i_1} is found, we extrapolate function ϕ on set $\mathrm{Dom}(\phi) \cup \{i_1\}$:

$$\mathrm{Dom}(\phi) = \mathrm{Dom}(\phi) \cup \{i_1\},\ \phi(i_1) = \mu_{i_1}$$

After that, we move to the next $i_1 \in [n] - \mathrm{Dom}(\phi)$.

Because $b^{i_0j_0}_{\mu_0\nu_0} = true$ is a noise, all these "random walks" will lead into dead ends, i.e. we will find that for some current $i_1 \in [n] - \mathrm{Dom}(\phi) \neq \emptyset$

$$\bigvee_{\mu\in[m_{i_1}]} b^{i_1i_1}_{\mu\mu} \wedge \bigwedge_{i\in\mathrm{Dom}(\phi)} b^{ii_1}_{\phi(i)\mu} = false$$

- we would extrapolate our partial index function ϕ to a total index function satisfying functional equation 1.3 what is impossible, otherwise.

Exercise 2.1 shows that functional equation 1.3 can be adequately encoded in some intrinsic semi-global characteristics of the figure depicted on compatibility matrix 1.1: global in m but not in n. Then, the compatibility-matrix encoding of the solution grids will be something like Gröbner bases.

Let's select $k \in [n]$. And let Q^k_n be the set of all k-combinations of n. Let's arbitrarily enumerate set Q^k_n:

$$Q^k_n = \{q_1, q_2, \cdots, q_{C^k_n}\} \qquad (2.1)$$

And let's arbitrarily enumerate each set $q_\alpha \in Q_n^k$:

$$q_\alpha = \{i_1^\alpha, i_2^\alpha, \ldots, i_k^\alpha\} \subseteq [n], \;\; \alpha \in [\mathrm{C}_n^k] \tag{2.2}$$

For each k-combination $q_\alpha \in Q_n^k$, let P_α be the following Cartesian product:

$$P_\alpha = [m_{i_1^\alpha}] \times [m_{i_2^\alpha}] \times \ldots \times [m_{i_k^\alpha}]$$

And let's arbitrarily enumerate each set P_α:

$$P_\alpha = \{p_1^\alpha, p_2^\alpha, \ldots, p^\alpha_{\prod_{\iota\in[k]} m_{i_\iota^\alpha}}\}, \;\; \alpha \in [\mathrm{C}_n^k] \tag{2.3}$$

- where $p_\beta^\alpha = (\mu_1^\beta, \mu_2^\beta, \ldots, \mu_k^\beta)$ is a k-tuple from set P_α, $\beta \in [\prod_{\iota\in[k]} m_{i_\iota^\alpha}]$.

And now, let's compute the following compatibility matrix from the given compatibility matrix 1.1:

$$R(k) = (R_{\alpha_1\alpha_2}(k))_{|Q_n^k|\times|Q_n^k|} = ((r_{\beta_1\beta_2}^{\alpha_1\alpha_2})_{|P_{\alpha_1}|\times|P_{\alpha_2}|})_{|Q_n^k|\times|Q_n^k|} \tag{2.4}$$

- where the matrix's components are

$$\begin{aligned} r_{\beta_1\beta_2}^{\alpha_1\alpha_2} = \; & (\alpha_1 \neq \alpha_2 \vee \textstyle\bigwedge_{(i,\mu),(j,\nu)\in\{(i_1^{\alpha_1},\mu_1^{\beta_1}),\ldots,(i_k^{\alpha_1},\mu_k^{\beta_1})\}} b_{\mu\nu}^{ij}) \\ \wedge \; & (\alpha_1 = \alpha_2 \vee \textstyle\bigwedge_{i_{\iota_1}^{\alpha_1}=i_{\iota_2}^{\alpha_2}} (\mu_{\iota_1}^{\beta_1} = \mu_{\iota_2}^{\beta_2})) \end{aligned} \tag{2.5}$$

- where $b_{\mu\nu}^{ij}$ are the components of the given compatibility matrix 1.1.

Exercise 2.2. Let's compute compatibility matrix 2.4 for the compatibility matrix from Exercise 1.3 and values $k = 1, 2, 3$.

For $k = 1$: $Q_3^1 = \{q_1 = (i_1^1 = 1), q_2 = (i_1^2 = 2), q_3 = (i_1^3 = 3)\}$; $P_1 = \{p_1 = (\mu_1^1 = 1), p_2 = (\mu_1^2 = 2)\}$, $P_2 = \{p_1 = (\mu_1^1 = 1), p_2 = (\mu_1^2 = 2)\}$, and $P_3 = \{p_1 = (\mu_1^1 = 1), p_2 = (\mu_1^2 = 2)\}$; and

$$R(1) = \left(\begin{array}{cc|cc|cc} 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \end{array}\right)$$

For $k = 2$: $Q_3^2 = \{q_1 = (i_1^1 = 1, i_2^1 = 2), q_2 = (i_1^2 = 1, i_2^2 = 3), q_3 = (i_1^3 = 2, i_2^3 = 3)\}$; $P_1 = P_2 = P_3 = \{p_1 = (\mu_1^1 = 1, \mu_2^1 = 1), p_2 = (\mu_1^2 = 1, \mu_2^2 =$

$2), p_3 = (\mu_1^3 = 2, \mu_2^3 = 1), p_4 = (\mu_1^4 = 2, \mu_2^4 = 2)\}$; and

$$R(2) = \left(\begin{array}{cccc|cccc|cccc}
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
\hline
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1
\end{array}\right),$$

And for $k = 3$: $Q_3^3 = \{q_1 = (i_1^1 = 1, i_2^1 = 2, i_3^1 = 3)\}$; $P_1 = \{p_1 = (\mu_1^1 = 1, \mu_2^1 = 1, \mu_3^1 = 1), p_2 = (\mu_1^2 = 1, \mu_2^2 = 1, \mu_3^2 = 2), p_3 = (\mu_1^3 = 1, \mu_2^3 = 2, \mu_3^3 = 1), p_4 = (\mu_1^4 = 1, \mu_2^4 = 2, \mu_3^4 = 2), p_5 = (\mu_1^5 = 2, \mu_2^5 = 1, \mu_3^5 = 1), p_6 = (\mu_1^6 = 2, \mu_2^6 = 1, \mu_3^6 = 2), p_7 = (\mu_1^7 = 2, \mu_2^7 = 2, \mu_3^7 = 1), p_8 = (\mu_1^8 = 2, \mu_2^8 = 2, \mu_3^8 = 2)\}$; and $R(3) = (0)_{8\times 8}$.

Basically, when $k \geq 2$, the diagonal components 2.5 encode the components of the given compatibility matrix 1.1 while the off-diagonal components 2.5 encode the box structure of the given compatibility matrix 1.1.

Lemma 2.3. *When $k \geq 2$, solution grids in compatibility matrix 2.4 and solution grids in compatibility matrix 1.1 are in one-to-one relation.*

Proof. Let ϕ be the index function of a solution grid in compatibility matrix 1.1. Then, for any k, the following function ψ is the index function of a solution grid in compatibility matrix 2.4:

$$\psi : \alpha \in [\|Q_n^k\|] \mapsto \beta \in [\|P_\alpha\|]$$

- where $\beta = \psi(\alpha)$ is such that $p_\beta^\alpha = (\phi(i_1^\alpha), \phi(i_2^\alpha), \ldots, \phi(i_k^\alpha))$. And, for any k, matching $\phi \to \psi$ of these index functions is an injection.

Let ψ be the index function of a solution grid in compatibility matrix 2.4. Let $\alpha \in [\|Q_n^k\|]$. Then, $\beta = \psi(\alpha) \in [\|P_\alpha\|]$. Let $q_\alpha = (i_1^\alpha, i_2^\alpha, \ldots, i_k^\alpha) \in Q_n^k$ and $p_\beta^\alpha = (\mu_1^\beta, \mu_2^\beta, \ldots, \mu_k^\beta) \in P_\alpha$. Due to functional equation 1.3,

$$r_{\psi(\alpha_1)\psi(\alpha_2)}^{\alpha_1\alpha_2} \equiv true$$

Then, due to the second conjunct on the right side of equalities 2.5, the following function ϕ is an one-meaning function, and it is an index function of compatibility matrix 1.1:

$$\phi : i_\iota^\alpha \in [n] \mapsto \mu_\iota^\beta \in [m_{i_\iota}]$$

- where $\alpha \in [\mathrm{C}^k_n]$, $\beta = \psi(\alpha)$, and $\iota \in [k]$. Matching $\psi \rightarrow \phi$ is an injection. And, due to the first conjunct on the right side of equalities 2.5,

$$\bigwedge_{q_\alpha \in Q^k_n} \bigwedge_{i,j \in [q_\alpha]} b^{ij}_{\phi(i)\phi(j)} = true$$

Then, when $k \geq 2$, index function ϕ is a solution of functional equation 1.3 for compatibility matrix 1.1: $b^{ij}_{\phi(i)\phi(j)} \equiv true,\ i,j \in [n]$. □

Due to Lemma 2.3, compatibility matrix 2.4 is a compatibility-matrix encoding of the compatibility matrix 1.1 when $k \geq 2$. Let's deplete compatibility matrix 2.4 with the following *zero propagation method*: we loop through compatibility boxes $R_{\alpha_1\alpha_2(k)}$ ($\alpha_1, \alpha_2 \in [\mathrm{C}^k_n]$) of compatibility matrix $R(k)$ and iterate the rows and columns in the current box; when the current row/column completely consists of the $false$-components, we propagate that value $false$ on the whole matrix $R(k)$ in the direction of the current row/column; and we continue while matrix $R(k)$ gets finalized.

Exercise 2.4. The zero propagation method transforms compatibility matrix $R(2)$ from Exercise 2.2 into the following compatibility matrix:

$$R'(2) = \left(\begin{array}{cccc|cccc|cccc}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right)$$

Let's notice, there still is noise in compatibility matrix $R'(2)$.

Basically, the zero propagation method realizes the light-ray notion. And, as in the descriptive geometry itself, it often fails to produce the general solution because of the geometrical features of the depicted figure such like concaveness. Nevertheless, we would like this method to work and transform compatibility matrix 2.4 into its general solution.

Let's restrict depletion in the zero propagation method to the diagonal compatibility boxes of matrix 2.4. Then, such restricted zero-propagation method can be expressed by the following system of material implications[3]:

$$(2.6)\qquad \begin{cases} 1) & \chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} \Rightarrow r^{\alpha_1\alpha_1}_{\beta_1\beta_1} \\ 2) & \chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} \Rightarrow \bigvee_{\{\beta | r^{\alpha_1\alpha}_{\beta_1\beta}=true\}} \chi^{\alpha\alpha}_{\beta\beta},\ \alpha \in [\mathrm{C}^k_n],\ \alpha \neq \alpha_1 \\ 3) & \quad true \Rightarrow \bigvee_{\beta\in[P_{\alpha_1}]} \chi^{\alpha_1\alpha_1}_{\beta\beta} \end{cases}$$

- where $\alpha_1 \in [\|Q^k_n\|] = [\mathrm{C}^k_n]$, $\beta_1 \in [\|P_{\alpha_1}\|] = [\prod_{i\in[k]} m_{i^{\alpha_1}}]$, and variables $\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1}$ are the unknown final values of the appropriate diagonal components.

Exercise 2.5. (1) Replacement of all implications in system 2.15 by their CNF-expressions[4] will transform system 2.6 into a dual Horn SAT instance. (2) System 2.6 is monotone: let $R_1(k)$ and $R_2(k)$ be two matrices 2.4, and let[5] $R_1(k) \Rightarrow R_2(k)$; then any solution of system 2.6 for matrix $R_1(k)$ is a solution of the system for matrix $R_2(k)$. (3) Set of solutions of system 2.6 is closed under disjunction[6]. (4) Material implication is a partial order on the set of all solutions of system 2.6. (5) If system 2.6 is consistent (it has solutions, i.e. such true assignments to variables $\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1}$ which satisfy all implications in the system), there is unique *maximal solution* which equals the disjunction of all solutions. (6) And that maximal solution can be obtained with the unit propagation method[7].

Minimal solution of system 2.6 is any true assignment λ,

$$(2.7)\qquad \chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} = \lambda^{\alpha_1\alpha_1}_{\beta_1\beta_1} \in \{false, true\},$$

which satisfies all implications in system 2.6 and in which the number of values *true* equals C^k_n:

$$(2.8)\qquad \sum_{\lambda^{\alpha_1\alpha_1}_{\beta_1\beta_1}} 1 = \mathrm{C}^k_n$$

[3] The second group of these implications is present only when $k < n$.

[4] $(x \Rightarrow y) \Leftrightarrow (\bar{x} \vee y)$

[5] For two Boolean matrices (a_{ij}) and (b_{ij}), $(a_{ij}) \Rightarrow (b_{ij})$ if $a_{ij} \Rightarrow b_{ij}$ for all i and j. Material implication is a partial order on the set of all Boolean matrices.

[6] If true assignments

$$\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} = \eta^{\alpha_1\alpha_1}_{\beta_1\beta_1} \in \{false, true\} \text{ and } \chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} = \theta^{\alpha_1\alpha_1}_{\beta_1\beta_1} \in \{false, true\}$$

are the solutions of system 2.6, then true assignment

$$\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} = \eta^{\alpha_1\alpha_1}_{\beta_1\beta_1} \vee \theta^{\alpha_1\alpha_1}_{\beta_1\beta_1}$$

is a solution of system 2.6 as well.

[7] For dual Horn CNF, the unit propagation method is as follows: for every each negative single-literal clause $\neg c$ aka unit, we remove from the CNF all literals c and all clauses containing literal $\neg c$; in this way, we propagate the units while there are any; the CNF is unsatisfiable iff this propagation will eliminate all literals from a clause; otherwise, when the propagation halts, we assign value *false* to all those variables which created the units, and we assign value *true* to the rest variables.

Due to the third group of implications in system 2.6, the minimal solutions of system 2.6 are in one-to-one relation with the solution grids in compatibility matrix 2.4 ($\alpha_1 \in [|Q_n^k|] = \mathrm{C}_n^k$ and $\beta_1 \in [|P_{\alpha_1}|]$):

$$\lambda_{\beta_1\beta_1}^{\alpha_1\alpha_1} = (\psi(\alpha_1) = \beta_1) = \bigwedge_{\iota\in[k]} (\phi(i_\iota^{\alpha_1}) = \mu_\iota^{\beta_1}) \tag{2.9}$$

- where ψ and ϕ are index functions of the appropriate solution grids in compatibility matrices 2.4 and 1.1. Therefore, we have one-one reduced the solution grids problem to the minimal solutions problem.

In the spirit of Ramsey theory, the following mini-max (m is maximum 1.2) is called the *uniform threshold* for compatibility matrix 2.4:

$$\kappa_1 = \min\{n, m+1\} \tag{2.10}$$

Lemma 2.6. *For any $k \geq \kappa_1$, any solution of system 2.6 for compatibility matrix 2.4 is a disjunction the system's minimal solutions. Particularly, the system is consistent for $k \geq \kappa_1$ iff its minimal solutions do exist.*

Proof. This lemma hods in the case when system 2.6 is inconsistent because there are not any solutions of the system in this case. To prove this lemma for consistent system 2.6, let's use mathematical induction over n.

For $n = k$, compatibility matrix 2.4 has only one compatibility box which is a diagonal Boolean matrix. So, this lemma obviously holds in this case.

Assuming that this lemma holds for some $n = l \geq k$, let's prove it for

$$n = l + 1 > k$$

Let σ be our system 2.6, and let σ' be the following subsystem of σ:

$$\sigma' : \; q_{\alpha}, q_{\alpha_1} \in Q_l^k$$

System σ' is system 2.6 for the $l \times l$ upper-left box corner B' of compatibility matrix 1.1. B' is a compatibility matrix on its own, and $n = l$ in B'. Therefore, this lemma holds for σ' due to our induction hypothesis.

Let R' be the submatrix of matrix 2.4 appropriate to subsystem σ'. And let Λ be the set of all minimal solutions of σ' (it is a finite set, $|\Lambda| \leq m^l$):

$$\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_{|\Lambda|}\}$$

- where each $\lambda_{*\in[|\Lambda|]}$ is a true assignment 2.7/2.8 satisfying subsystem σ' ($\sum_{\lambda_{\beta_1\beta_1}^{\alpha_1\alpha_1}} 1 = \mathrm{C}_l^k$). Then, due to our induction hypothesis, any solution $\chi_{\beta_1\beta_1}^{\alpha_1\alpha_1} = \eta_{\beta_1\beta_1}^{\alpha_1\alpha_1}$ of σ' (any true assignment $\chi_{\beta_1\beta_1}^{\alpha_1\alpha_1} = \eta_{\beta_1\beta_1}^{\alpha_1\alpha_1}$ which satisfies all implications in σ') is a disjunction of some true assignments from set Λ:

$$\eta_{\beta_1\beta_1}^{\alpha_1\alpha_1} = \bigvee_{\lambda\in\eta\subseteq\Lambda} \lambda_{\beta_1\beta_1}^{\alpha_1\alpha_1}$$

- where η is the appropriate subset of Λ, appropriate to true assignment $\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} = \eta^{\alpha_1\alpha_1}_{\beta_1\beta_1}$. Let's notice, this decomposition is not unique, in general.
Now, let the following true assignment be a solution of system σ:

$$\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} = \theta^{\alpha_1\alpha_1}_{\beta_1\beta_1} \in \{false, true\}, \ \alpha_1 \in [\mathrm{C}^k_{l+1}]$$

Then, subset $q_{\alpha_1} \in Q^k_l$ of these assignments is a solution of subsystem σ'. Then, there is subset $\theta \subseteq \Lambda$ such that

$$\theta^{\alpha_1\alpha_1}_{\beta_1\beta_1} = \bigvee_{\lambda\in\theta} \lambda^{\alpha_1\alpha_1}_{\beta_1\beta_1}, \ q_{\alpha_1} \in Q^k_l$$

Then, due to one-to-one relation 2.9,

$$\theta^{\alpha_1\alpha_1}_{\beta_1\beta_1} = \bigvee_{\lambda\in\theta} (\psi_\lambda(\alpha_1) = \beta_1) = \bigvee_{\lambda\in\theta} \ \bigwedge_{\iota\in[k]} (\phi_\lambda(i^{\alpha_1}_\iota) = \mu^{\beta_1}_\iota), \ q_{\alpha_1} \in Q^k_l$$

- where ψ_λ and ϕ_λ are index functions 2.9 of the solution grids in compatibility matrices R' and B' appropriate to the minimal solutions $\lambda \in \theta$ of subsystem σ' of system σ. Then, due to the second and first groups of implications in system σ (where $n = l + 1 \geq 2$), for any minimal solution $\lambda \in \theta$, k-combination $q_{\alpha_1} \in Q^k_l$, and k-combination $q_\alpha \in Q^k_{l+1} - Q^k_l \neq \emptyset$

$$true = \theta^{\alpha_1\alpha_1}_{\psi_\lambda(\alpha_1)\psi_\lambda(\alpha_1)} \ \Rightarrow \bigvee_{\{\beta | r^{\alpha_1\alpha}_{\psi_\lambda(\alpha_1)\beta} = true\}} \theta^{\alpha\alpha}_{\beta\beta} \ \Rightarrow \bigvee_{\{\beta | r^{\alpha_1\alpha}_{\psi_\lambda(\alpha_1)\beta} = true\}} r^{\alpha\alpha}_{\beta\beta}$$

Then (see Exercise 2.1), because $k \geq \kappa_1$, each partial index function ϕ_λ, $\lambda \in \theta$, can be extrapolated from set $[l]$ on set $[l+1]$ for each $\theta^{\alpha_1\alpha_1}_{\beta_1\beta_1} = true$, $q_{\alpha_1} \in Q^k_{l+1} - Q^k_l$, due to equalities 2.5 and the rest implications in system σ. Each of these extrapolations satisfies functional equation 1.3 for the given compatibility matrix 1.1 and creates a minimal solution 2.9 of σ. And solution $\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} = \theta^{\alpha_1\alpha_1}_{\beta_1\beta_1}$ of σ is disjunction of all those minimal solutions. Therefore, this lemma holds for $n = l + 1$. □

System 2.6 functions as follows: the first group of implications is an input; the third group of implications filters the input against a rule; and the third group of implications tests the result.

Theorem 2.7. *For compatibility matrix 1.1:*
1). Let's compute compatibility matrix 2.4 for any $k \geq \kappa_1$;
2). Let's compile dual Horn CNF (it will be always satisfiable)

$$h_2 = \bigwedge\nolimits_{r^{\alpha_1\alpha_1}_{\beta_1\beta_1}} \neg\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} \ \wedge \ \bigwedge\nolimits_{\alpha_1,\alpha\in[|Q^k_n|],\alpha\neq\alpha_1} (\neg\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} \vee \bigvee\nolimits_{\{\beta | r^{\alpha_1\alpha}_{\beta_1\beta}\}} \chi^{\alpha\alpha}_{\beta\beta})$$

3). Let's deploy unit propagation method (see Exercise 2.5) to formula h_2 and compute the appropriate satisfying true assignment

$$\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} = \eta^{\alpha_1\alpha_1}_{\beta_1\beta_1} \in \{false, true\};$$

4). Let's compile Horn CNF (it will be always satisfiable)

$$\begin{array}{rl} h_1 = & \bigwedge_{\eta^{\alpha_1\alpha_1}_{\beta_1\beta_1}} \bigwedge_{(i,\mu),(j,\nu)\in\{(i^{\alpha_1}_1,\mu^{\beta_1}_1),\ldots,(i^{\alpha_1}_k,\mu^{\beta_1}_k)\}} \xi^{ij}_{\mu\nu} \\ \wedge & \bigwedge_{\neg\eta^{\alpha_1\alpha_1}_{\beta_1\beta_1}} \cdot \bigvee_{(i,\mu),(j,\nu)\in\{(i^{\alpha_1}_1,\mu^{\beta_1}_1),\ldots,(i^{\alpha_1}_k,\mu^{\beta_1}_k)\}} \neg\xi^{ij}_{\mu\nu} \end{array} \;;$$

5). Let's deploy unit propagation method[8] to formula h_1 and compute the appropriate satisfying true assignment

$$\xi^{ij}_{\mu\nu} = \theta^{ij}_{\mu\nu} \in \{false, true\}.$$

The general solution of compatibility matrix 1.1 equals $((\theta^{ij}_{\mu\nu})_{m_i\times m_j})_{n\times n}$.

Proof. It is a direct consequence of Lemma 2.3, Lemma 2.6 (we just removed the third group of implications from system 2.6), and equalities 2.5. □

As a mater of fact, Theorem 2.7 is an analog of Gauss exclusions method. The theorem resolves the search problem for the compatibility-matrix encoded problem. If we have just decision problem, we may rid of the backfeeds, and everything gets simplified a little. System 2.6 becomes reduced:

$$(2.11) \qquad \left\{ \begin{array}{llll} 1) & \chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} & \Rightarrow & r^{\alpha_1\alpha_1}_{\beta_1\beta_1} \\ 2) & \chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} & \Rightarrow & \bigvee_{\{\beta | r^{\alpha_1\alpha}_{\beta_1\beta}=true\}} \chi^{\alpha\alpha}_{\beta\beta},\ \alpha\in[\mathrm{C}^k_n],\ \alpha>\alpha_1 \\ 3) & true & \Rightarrow & \bigvee_{\beta\in[P_1]} \chi^{\alpha_1\alpha_1}_{\beta\beta} \end{array} \right.$$

This system is consistent iff the following dual Horn CNF is satisfiable:

$$h_3 = \bigwedge_{r^{\alpha_1\alpha_1}_{\beta_1\beta_1}} \neg\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} \wedge \bigwedge_{\alpha_1,\alpha\in[\|Q^k_n\|],\alpha>\alpha_1} (\neg\chi^{\alpha_1\alpha_1}_{\beta_1\beta_1} \vee \bigvee_{\{\beta | r^{\alpha_1\alpha}_{\beta_1\beta}\}} \chi^{\alpha\alpha}_{\beta\beta}) \wedge \bigvee_{\beta\in[P_1]} \chi^{\alpha_1\alpha_1}_{\beta\beta}$$

Formula h_3 is satisfiable iff there are solution grids in compatibility matrix 1.1. And satisfiability of formula h_3 can be tested with the unit propagation method. Computational complexity of the unit propagation method is linear in the formula size which is $O(m^k(\mathrm{C}^k_n)^2)$. We can control the computational complexity to some extend with k, $\kappa_1 \leq k \leq n$. And the minimal value of the computational complexity is $O(m^{\kappa_1}(\mathrm{C}^{\kappa_1}_n)^2)$ when $k = \kappa_1$.

Substitution $\chi^{\alpha\alpha}_{\beta\beta} = \neg\zeta^{\alpha\alpha}_{\beta\beta}$ transforms system 2.11 into the following monotone circuit for the decision problems:

$$(2.12) \qquad \left\{ \begin{array}{llll} 1) & \zeta^{\alpha_1\alpha_1}_{\beta_1\beta_1} & \Leftarrow & \neg r^{\alpha_1\alpha_1}_{\beta_1\beta_1} \\ 2) & \zeta^{\alpha_1\alpha_1}_{\beta_1\beta_1} & \Leftarrow & \bigvee_{\alpha\in[\mathrm{C}^k_n],\ \alpha>\alpha_1} \bigwedge_{\{\beta | r^{\alpha_1\alpha}_{\beta_1\beta}=true\}} \zeta^{\alpha\alpha}_{\beta\beta} \\ 3) & \zeta_0 & \Leftarrow & \bigwedge_{\beta\in[P_1]} \zeta^{\alpha_1\alpha_1}_{\beta\beta} \end{array} \right.$$

[8] For Horn CNF, the unit propagation method is as follows: for every each positive single-literal clause c aka unit, we remove from the CNF all literals $\neg c$ and all clauses containing literal c; in this way, we propagate the units while there are any; the CNF is unsatisfiable iff this propagation will eliminate all literals from a clause; otherwise, when the propagation halts, we assign value *true* to all those variables which created the units, and we assign value *false* to the rest variables.

In this circuit, values $\neg r^{\alpha_1\alpha_1}_{\beta_1\beta_1}$ are an input, and value ζ_0 is output. The output equals *false* iff the compatibility-matrix encoded problem is consistent. The circuit's depth and width are $O(m^k \mathrm{C}^k_n)$. The circuit's depths and width can be controlled with k, and their minimum is $O(m^{\kappa_1}\mathrm{C}^{\kappa_1}_n)$.

Strictly speaking, circuit 2.12 is not uniform. Yet, it can be fixed with the protocoling of enumerations 2.1, 2.2, and 2.3 for different box structures of compatibility matrix 1.1. And the number of the different box structures can be reduced with, for example, the padding of the smaller compatibility boxes by the rows/columns entirely filled with *false*. Then, systems 2.6, 2.11, and 2.12 can be tabulated and wired. It would reduce the compatibility matrix method to the compatibility-matrix encoding.

Another approach to the uniformity is to rid of the references to the components of compatibility matrix 2.4 in the indices of system 2.6 and its derivatives. It can be accomplished with, for example, the replacement of compatibility matrix 2.4 by the following compatibility matrix:

$$S(k) = (S_{\alpha_1\alpha_2}(k))_{|Q^k_n|\times|Q^k_n|} = ((s^{\alpha_1\alpha_2}_{\beta_1\beta_2})_{|P_{\alpha_1}|\times|P_{\alpha_2}|})_{|Q^k_n|\times|Q^k_n|} \tag{2.13}$$

- where

$$s^{\alpha_1\alpha_2}_{\beta_1\beta_2} = \bigwedge_{(i,\mu),(j,\nu)\in\{(i^{\alpha_1}_1,\mu^{\beta_1}_1),\ldots,(i^{\alpha_1}_k,\mu^{\beta_1}_k)\}\cup\{(i^{\alpha_2}_1,\mu^{\beta_2}_1),\ldots,(i^{\alpha_2}_k,\mu^{\beta_2}_k)\}} b^{ij}_{\mu\nu} \tag{2.14}$$

- where $b^{ij}_{\mu\nu}$ are the components of the given compatibility matrix 1.1. And with the replacement of system 2.6 by the following system:

$$\begin{cases} 1) & \chi^{\alpha_1\alpha_2}_{\beta_1\beta_2} & \Rightarrow & s^{\alpha_1\alpha_2}_{\beta_1\beta_2} \\ 2) & \chi^{\alpha_1\alpha_2}_{\beta_1\beta_2} & \Rightarrow & \bigvee_{\beta\in[|P_\alpha|]} \chi^{\alpha_1\alpha}_{\beta_1\beta},\ \alpha\neq\alpha_2 \\ & \chi^{\alpha_1\alpha_2}_{\beta_1\beta_2} & \Rightarrow & \bigvee_{\beta\in[|P_\alpha|]} \chi^{\alpha\alpha_2}_{\beta\beta_2},\ \alpha\neq\alpha_1 \\ 3) & \mathit{true} & \Rightarrow & \bigvee_{\beta\in[|P_{\alpha_1}|]} \chi^{\alpha_1\alpha_1}_{\beta\beta} \end{cases} \tag{2.15}$$

It will even reduce uniform threshold 2.10: everything above can be edited for compatibility matrix 2.13 and system 2.15 in assumption $k \geq \kappa_0$, where

$$\kappa_0 = \min\{n, m\} \tag{2.16}$$

Still, matrix 2.4 and its derivatives seem to be easier for the preprocessing.

3. Compatibility boxes

Solution grids in compatibility matrix 1.1 are in one-to-one relation with the true assignments satisfying the following Boolean formula[9]:

$$g_1 = \bigwedge_{i\in[n]} \bigoplus_{\mu\in[m_i]} \chi^i_\mu \wedge \bigwedge_{\neg b^{ij}_{\mu\nu}} (\neg\chi^i_\mu \vee \neg\chi^j_\nu) \tag{3.1}$$

[9] "$\oplus$" is the XOR operator.

The one-to-one relation is as follows:

$$\chi^i_\mu = (\phi(i) = \mu),\ \ i \in [n], \mu \in [m_i]$$

- where ϕ is index function of a solution grid in compatibility matrix 1.1. The $O(m^2n)$-time replacement of all XOR operators in formula g_1 by their CNF-expressions[10] will one-one reduce formula g_1 to CF

$$g_2 = \bigwedge_{i\in[n]} \ \bigvee_{\mu\in[m_i]} \ \chi^i_\mu \ \wedge \ \bigwedge_{\neg b^{ij}_{\mu\nu}} (\neg\chi^i_\mu \vee \neg\chi^j_\nu) \tag{3.2}$$

And Cook-Karp equivalence transformations[11] [9, 10] will one-one reduce CF g_2 to a 3CF. In the 3CF, every couple of the slack literals ξ and $\neg\xi$ can be replaced by two variables $\zeta_1 = \xi$, $\zeta_2 = \neg\xi$, and constrain

$$(\zeta_1 \oplus \zeta_2 = true) \ \Leftrightarrow \ ((\zeta_1 \vee \zeta_2) \wedge (\neg\zeta_1 \vee \neg\zeta_2) = true)$$

Because of the XOR operators in g_1, these constrains can be reduced:

$$(\zeta_1 \oplus \zeta_2 = true)|_{g_2} \ \Leftrightarrow \ (\neg\zeta_1 \vee \neg\zeta_2 = true)$$

Substitutions $\zeta_1 = \xi$, $\zeta_2 = \neg\xi$ in the 3CF and adding clauses $\neg\zeta_1 \vee \neg\zeta_2$ to it will one-one reduce that 3CF into formula 3.2 for a compatibility matrix with the compatibility boxes of size 3 x 3 or less. When needed, the compatibility boxes whose size is less than 3 x 3 can be always padded[12]. Therefore, the following analog of Cook's 3SAT theorem [9] holds:

Theorem 3.1. *Compatibility matrix 1.1 is $O(m^2n^2)$-time one-one reducible to a $O(mn \times mn)$ compatibility matrix with the* 3 x 3 *compatibility boxes.*

The $O(m^2n^2)$-time one-one reduction of the given compatibility matrix 1.1 to a compatibility matrix with the 3 x 3 compatibility boxes may be included in the compatibility matrix method in the preprocessing. Then, the tabulation/wiring of enumerations 2.1, 2.2, and 2.3 for the 3 x 3 compatibility boxes will reduce the time-complexity of the compatibility matrix method to $O(m^4n^4)$ and the space-complexity to $O(m^8n^8)$: the $O(m^4n^4)$-depth/width circuit 2.12 performs parallel $O(m^4n^4)$-time computing.

Any particular protocol for enumerations 2.1, 2.2, and 2.3 for compatibility matrices with the 3 x 3 compatibility boxes will make the sets of the appropriate systems 2.6 and 2.15 (and the sets of their derivatives) the sparse sets. And computation of the implicants (the right sides) in the first groups of implications in the systems will be Karp reduction of the problems allowing the compatibility-matrix encoding to those sparse sets.

[10] $x \oplus y \oplus \ldots \oplus z \Leftrightarrow (x \vee y \vee \ldots \vee z) \wedge (\bar{x} \vee \bar{y}) \wedge \ldots (\bar{x} \vee \bar{z}) \wedge \ldots \wedge (\bar{y} \vee \bar{z}) \wedge \ldots$

[11] $x \vee y \vee z \vee \ldots \Leftrightarrow (x \vee y \vee \neg\xi) \wedge (\xi \vee z \vee \ldots) \Leftrightarrow \ldots$

[12] For example, $x \Leftrightarrow (x \vee \neg\xi) \wedge (x \vee \xi) \Leftrightarrow \ldots$

4. Symmetries

Compatibility-matrix encoding allows another interpretation which may well serve as the encoding recipe: there is a *placeholder* for the solution of the given problem; the placeholder is partitioned in n parts which may overlap; there are options for each of the parts - set O_i for part i, $m_i = |O_i|$; there are n^2 2-ary relations $\rho_{ij} \subseteq O_i \times O_j$ among the options - the diagonal relations ρ_{ii} are the inclusion relations, and the off-diagonal relations ρ_{ij}, $i \neq j$, are the *compatibility relations*; the parts and their options are enumerated; and compatibility boxes B_{ij} are the appropriate graphics of relations ρ_{ij}. Then, the problem itself is a system of relations:

$$(x_i, x_j) \in \rho_{ij},\ i, j \in [n] \tag{4.1}$$

- where variables $x_i \in O_i$ are the unknown options for the placeholder's parts. And the problems allowing the compatibility-matrix encoding are the only problems allowing binary decomposition 4.1.

Exercise 4.1. Along with compatibility relations we could use the *contradictory relations.* Basically, the *contradictory matrix* is negation of the compatibility matrix:

$$\neg B = (\neg B_{ij})_{n \times n} = ((\neg b^{ij}_{\mu\nu})_{m_i \times m_j})_{n \times n}$$

The contradictory and compatibility matrices are De Morgan dual.

Relabeling of the placeholder' parts and the parts' options, remodeling of the placeholder and its partition, and reducibility among the problems allowing the compatibility-matrix encoding all that may be seen as symmetries in the set of all compatibility matrices: the relabeling transforms the given compatibility matrix into a congruent compatibility matrix[13]; the remodeling is covered to some extend by Theorem 3.1; and partition of the problems on consistent and inconsistent is covered to some extend by Theorems 3.1 and 2.7 - computational complexity of the partition's projection is polynomial in the compatibility matrix's size. Let's see other partitions.

Theorem 2.7 gives the general solution of compatibility matrix 1.1. Then, the following Boolean equation gives an exterior rectangular estimation of the relation created by the problem encoded by compatibility matrix 1.1:

$$\bigwedge_{i \in [n]} \bigvee_{\theta^{ii}_{\mu\mu}} \chi^i_\mu = true \tag{4.2}$$

And this estimation is exact: Boolean equation 4.2 is consistent iff the encoded problem is consistent. The vertices of rectangle 4.2 are solutions

[13] Two square box matrices are congruent if they can be transformed one into other with the permutations of the same rows and columns of boxes and with the permutations of the same rows and columns of components inside of the rows and columns of boxes.

of the following Boolean equation:

$$\bigwedge_{i\in[n]} \bigoplus_{\theta^{ii}_{\mu\mu}} \chi^i_\mu = true \tag{4.3}$$

This equation is the affine case in Schafer's classification [10]. And its solutions can be expressed either by system of linear inequalities or even by system of linear equations in Euclidean space $\mathbf{R}^{\sum_{i\in[n]} m_i} = \mathbf{R}^{O(mn)}$:

$$\left\{ \begin{array}{rl} \sum_{\mu\in[m_i]} x^i_\mu = 1, & i \in [n] \\ x^i_\mu = 0, & \neg\theta^{ii}_{\mu\mu} \\ 0 \leq x^i_\mu \leq 1 & \end{array} \right. \Leftrightarrow \left\{ \begin{array}{rl} \sum_{\mu\in[m_i]} x^i_\mu = 1, & i \in [n] \\ x^i_\mu = 0, & \neg\theta^{ii}_{\mu\mu} \\ & \end{array} \right. \tag{4.4}$$

Relabeling of the placeholder's parts and the parts' options in system 4.1 will rotate coordinates in this Euclidean space. And the asymmetric solution sets of these systems will "rotate" along with the coordinates. Therefore, systems 4.4 are asymmetric in the sense of Yannakakis' theorem [11].

With Theorem 3.1, system of linear equations 4.4 in the appropriate Euclidean space $\mathbf{R}^{O(mn)}$ can be obtained in the time polynomial in mn. The set of all such linear systems can be partitioned with the congruence of the systems' matrices - projection of this partition is linear in mn. And each of the equivalence classes can be presented by the system in which

$$3 \geq \sum_{\theta^{11}_{\mu\mu}} 1 \geq \sum_{\theta^{22}_{\mu\mu}} 1 \geq \ldots \geq \sum_{\theta^{O(mn)O(mn)}_{\mu\mu}} 1 \geq 0$$

Therefore, this partition of the problem allowing the compatibility-matrix encoding - the partition based on the exterior estimation 4.4 and relabeling of the placeholder's parts and the parts' options in system 4.1 - is another sparse set along with the sparse sets of Boolean systems 2.6, 2.15, and their derivatives which were described earlier at the end of Section 3.

Let's notice, the sparse sets described in Section 3 consist of the "universal algorithms" while the sparse set described in this section consists of the "encodings". In the other words, the difficulty of the problems allowing the compatibility matrix encoding can be dissolved by the "lucky choice" of the placeholder, the placeholder's partition, and the partition's labeling.

Exercise 4.2. Boolean equation 4.3 is Karp reducible to 2SAT through the replacement of the XOR operators by their CNF-expression and usual self-reducibility of the resulting CNF. Therefore, for the problems allowing the compatibility-matrix encoding, all cases in Schafer's classification are Karp reducible each to other except the case reserved for intractable problems, and that case lays outside of the compatibility matrix method's domain.

We have discussed the symmetry based on the exact exterior rectangular estimations 4.2 and 4.3. But, there are the similar interior estimations.

The solution grids in compatibility matrix 1.1 are in one-to-one relation with the true assignments satisfying Boolean formula 3.1. And formula 3.1 is self-reducible with the following equivalence transformations:

$$(x \oplus y \oplus \ldots \oplus z) \wedge (\bar{x} \vee X) \wedge (\bar{y} \vee Y) \wedge \ldots \wedge (\bar{z} \vee Z) \;\Leftrightarrow\; X \vee Y \vee \ldots \vee Z$$

- where variables $x, y, \ldots, z$ are independent and formulae $X, Y, \ldots, Z$ are independent of variables $x, y, \ldots, z$. These equivalence transformations will transform formula 3.1 into the following formula (computational complexity of this direct reduction will be exponential-time, in general):

$$(4.5) \qquad g_1{}' = \bigwedge_{i \in N \subseteq [n]} \Big(\bigoplus_{\mu \in [m_i]} \chi^i_\mu \wedge \bigwedge_{\mu \in M_i \subseteq [m_i]} \neg \chi^i_\mu \Big)$$

- where $N \neq \emptyset$ and some $M_i = [m_i]$ in the case of unsatisfiable formula 3.1.

Exercise 4.3. Formula 3.1 for the compatibility matrix from Exercise 1.3:

$$\begin{array}{rl}
g_1 = & (\chi^1_1 \oplus \chi^1_2) \wedge (\bar{\chi}^1_1 \vee \bar{\chi}^2_2) \wedge (\bar{\chi}^1_1 \vee \bar{\chi}^3_1) \wedge (\bar{\chi}^1_2 \vee \bar{\chi}^2_1) \wedge (\bar{\chi}^1_2 \vee \bar{\chi}^3_2) \\
\wedge & (\chi^2_1 \oplus \chi^2_2) \wedge (\bar{\chi}^2_1 \vee \bar{\chi}^3_2) \wedge (\bar{\chi}^2_2 \vee \bar{\chi}^3_1) \\
\wedge & (\chi^3_1 \oplus \chi^3_2) \\
\Leftrightarrow & (\bar{\chi}^2_2 \wedge \bar{\chi}^3_1 \vee \bar{\chi}^2_1 \wedge \bar{\chi}^3_2) \\
\wedge & (\chi^2_1 \oplus \chi^2_2) \wedge (\bar{\chi}^2_1 \vee \bar{\chi}^3_2) \wedge (\bar{\chi}^2_2 \vee \bar{\chi}^3_1) \\
\wedge & (\chi^3_1 \oplus \chi^3_2) \\
\wedge & (\chi^3_1 \oplus \chi^3_2) \\
\Leftrightarrow & (\bar{\chi}^2_2 \vee \bar{\chi}^2_1) \wedge (\bar{\chi}^3_1 \vee \bar{\chi}^2_1) \wedge (\bar{\chi}^2_2 \vee \bar{\chi}^3_2) \wedge (\bar{\chi}^3_1 \vee \bar{\chi}^3_2) \\
\wedge & (\chi^2_1 \oplus \chi^2_2) \wedge (\bar{\chi}^2_1 \vee \bar{\chi}^3_2) \wedge (\bar{\chi}^2_2 \vee \bar{\chi}^3_1) \\
\wedge & (\chi^3_1 \oplus \chi^3_2) \\
\hline
\Leftrightarrow & (\chi^2_1 \oplus \chi^2_2) \wedge (\bar{\chi}^2_1 \vee \bar{\chi}^3_2) \wedge (\bar{\chi}^2_2 \vee \bar{\chi}^3_1) \wedge (\bar{\chi}^3_1 \vee \bar{\chi}^2_1) \wedge (\bar{\chi}^2_2 \vee \bar{\chi}^3_2) \\
\wedge & (\chi^3_1 \oplus \chi^3_2) \\
\Leftrightarrow & (\bar{\chi}^3_2 \wedge \bar{\chi}^3_1 \vee \bar{\chi}^3_1 \wedge \bar{\chi}^3_2) \\
\wedge & (\chi^3_1 \oplus \chi^3_2) \\
\hline
\Leftrightarrow & (\chi^3_1 \oplus \chi^3_2) \wedge \bar{\chi}^3_1 \wedge \bar{\chi}^3_2 \;\; = g_1{}'
\end{array}$$

We see, $M_3 = [m_3] = \{1, 2\}$ in formula 4.5 for that compatibility matrix. In the case of a compatibility matrix with the larger compatibility boxes, the length of the newly created all-negative OR-clauses can increase at the beginning of the equivalence transformations. But, the length is bounded by the number of remaining XOR-clauses. So, it will decrease at the end.

Boolean equation $g_1{}' = \mathit{true}$ defines vertices of rectangle

$$\bigwedge_{i \in N} \; \bigvee_{\mu \in [m_i] - M_i} \chi^i_\mu = \mathit{true}$$

This dimensionally degenerated rectangle lays inside of the solution set of system 4.1. And this interior estimation is exact: there are solution grids in compatibility matrix 1.1 iff this rectangle is not empty. Thus, all above for exterior rectangle 4.2 holds for this interior rectangle.

Exercise 4.4. Formula 4.5 is not unique and depends on which XOR-clauses we choose to exclude from formula 3.1. Suppose, we have excluded the ith XOR-clause. Then,

$$\neg\chi^i_\mu \Leftarrow f^i_\mu,\ \mu \in [m_i]$$

- where each f^i_μ is an exponential-size all-positive DNF on the remaining variables. These implications create an AND-OR-NOT circuit of the linear depth. Unlike circuits 2.12, this family of circuits is nonuniform.

5. Similarity to quantum computer

System 4.1 reassembles quantum computer: variables x_i act like qubits, compatibility relations ρ_{ij} act like entanglement, and the system itself acts like superposition. And the following exact deterministic algorithm, which is called *depletion by multiplication*, reassembles Grover's algorithm [13].

Theorem 5.1. *For compatibility matrix 1.1:*
1). Let's compute compatibility matrix 2.13 - matrix $S(k)$;
2). Let's iterate compatibility boxes in compatibility matrix $S(k)$ and recompute the current box with formula[14]

$$S_{\alpha_1\alpha_2}(k) = \bigwedge_{\alpha=1}^{|Q^k_n|} S_{\alpha_1\alpha}(k) \cdot S_{\alpha\alpha_2}(k) \tag{5.1}$$

3). Let's loop these iterations until compatibility matrix $S(k)$ got finalized. The final value of compatibility matrix $S(k)$ is general solution of compatibility matrix 2.13 if $k \geq 0.5\kappa_0$, where κ_0 is uniform threshold 2.16.

Proof. Conjunction 5.1 depletes compatibility matrix $S(k)$ because the diagonal compatibility boxes are diagonal Boolean matrices. This algorithm will stop after $O((m^k \mathrm{C}^k_n)^2)$ loops because there are $(m^k \mathrm{C}^k_n)^2$ *true*-components in compatibility matrix $S(k)$ at most.

Due to the compatibility matrices' symmetry,

$$\bigwedge_{\alpha=1}^{|Q^k_n|} S_{\alpha_1\alpha}(k) \cdot S_{\alpha\alpha_2}(k) = \bigwedge_{\alpha=1}^{|Q^k_n|} \Big(\bigvee_{\beta\in|P_\alpha|} s^{\alpha_1\alpha}_{\beta_1\beta} \wedge s^{\alpha_2\alpha}_{\beta_2\beta}\Big)_{|P_{\alpha_1}|\times|P_{\alpha_2}|}$$

Therefore, solution grids in compatibility matrix 2.13 are invariants under transformation 5.1. On other hand, due to equalities 2.14, the solution grids

[14] We use the following Boolean matrix multiplication "$\cdot$":

$$(a_{i\mu}) \cdot (b_{\mu j}) = \Big(\bigvee_\mu a_{i\mu} \wedge b_{\mu j}\Big)$$

- where the number of columns in $(a_{i\mu})$ equals the number of rows in $(b_{\mu j})$. And formula 5.1 is the power of box matrix $S(k)$ in the sense of the Boolean box matrix multiplication.

in compatibility matrix 2.13 are in one-to-one relation with the solution grids in compatibility matrix 1.1:

$$\psi(\alpha) = \beta \Leftrightarrow \bigwedge_{\iota \in [k]} (\phi(i_\iota^\alpha) = \mu_\iota^\beta)$$

- where ψ and ϕ are index functions of the appropriate solution grids in compatibility matrices 2.13 and 1.1. Then (see Exercise 2.1), while compatibility matrix $S(k)$ for $k \geq \kappa_0/2$ contains noise, there always will be such indices $\alpha_1, \alpha_2, \alpha \in [\|Q_n^k\|] = [\mathrm{C}_n^k]$ that $s_{\beta_1\beta_2}^{\alpha_1\alpha_2} = true$ and

$$\bigvee_{\beta \in |P_\alpha|} s_{\beta_1\beta}^{\alpha_1\alpha} \wedge s_{\beta_2\beta}^{\alpha_2\alpha} \Rightarrow \bigvee_{\mu \in [m_{i_0}]} b_{\mu\mu}^{i_0 i_0} \wedge \bigwedge_{\iota \in [k]} b_{\mu_\iota \mu}^{i_\iota i_0} = false$$

for some $i_0 \in q_\alpha$, $i_1, i_2, \ldots, i_k \in q_{\alpha_1} \cup q_{\alpha_2}$, and the appropriate indices μ_ι, $\iota \in [k]$, from k-tuples $p_{\beta_1} \in P_{\alpha_1}$ and $p_{\beta_2} \in P_{\alpha_2}$. Therefore, when $k \geq \kappa_0/2$, the looping will continue while there is noise in matrix $S(k)$. □

Formula 5.1 is a definition of the power of box matrix $S(k)$ in the respect to the Boolean box matrix multiplication. Therefore, because of the conjunctions in the formula, the number of loops in the algorithm cannot be greater than the length of the longest path in the graph (with loops) whose adjacency matrix is compatibility matrix 2.13 after the replacement of *false* by 0 and *true* by 1. It is C_n^k-partite graph. So, the number of loops in the depletion by multiplication algorithm is $O(\mathrm{C}_n^k)$. And the time-complexity of the depletion by multiplication algorithm can be estimated as $O((m^k\mathrm{C}_n^k)^4)$ while its space-complexity is $O((m^k\mathrm{C}_n^k)^2)$, $k \geq \kappa_0/2$.

The depletion by multiplication is filtering of the compatibility boxes each against others. Before the filtering, we may use Theorem 3.1 and split the given compatibility matrix 1.1 into the $O(mn)$ compatibility boxes of size 3 × 3. It will make $\kappa_0 = 3$ and $k \geq 2 > 3/2$. Then, in full compliance with Baker, Gill, and Solovey's theorem [14], the number of loops in the depletion by multiplication will be controllable with k from $O(m^2n^2)$ when $k = 2$, up to $O(2^{mn})$ when $k \approx mn/2$, and down to $O(1)$ when $k \approx mn$.

The splitting of the compatibility matrix and selection of $k = 2$ will minimize the time-complexity of the depletion by multiplication algorithm, which will be $O(m^8n^8)$, and the space-complexity of the algorithm, which will be $O(m^4n^4)$. The time- and space-complexities swapped in the monotone circuit 2.11 and the depletion by multiplication "quantum algorithm." So, it seems, these approaches are on Pareto frontier of the computational complexity of the compatibility matrix method.

Indeed, the depletion by multiplication works as well for compatibility matrix 2.4 and $k \geq 0.5\kappa_1$, where κ_1 is uniform threshold 2.10.

Exercise 5.2. The depletion by multiplication works directly on compatibility matrix 1.1 when $m \leq 2$, where m is maximum 1.2. For example, for the compatibility matrix from Exercise 1.3,

$$B_{12} = B_{11} \cdot B_{12} \wedge B_{12} \cdot B_{22} \wedge B_{13} \cdot B_{32} =$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \wedge \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \wedge \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \wedge \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \wedge \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

There is no need to continue the depletion because it will just propagate this *false*-box B_{12} all over the rest of compatibility matrix. It is a *pattern of unsatisfiability* when formula 5.1 produces (and it will always do so sooner or later for the inconsistent problems) a compatibility box completely filled with *false*. When the pattern of unsatisfiability is detected, the depletion by multiplication may be stopped with the decision "No."

6. Application to the P vs NP problem

6.1. SAT. Let f be the given CNF:

$$f = \bigwedge_{i\in[n]} \bigvee_{\mu\in m_i} L^i_\mu, \; L^i_\mu \in \{x_\alpha, \neg x_\alpha \mid \alpha \in A\} \tag{6.1}$$

- where x_α are independent Boolean variables, and L^i_μ are the variables' literals. The problem is to decide whether or not f is satisfiable.

The following box matrix $((b^{ij}_{\mu\nu})_{m_i \times m_i})_{n\times n}$ is a compatibility matrix for SAT instance 6.1:

$$b^{ij}_{\mu\nu} = (i \neq j \vee \mu = \nu) \wedge (i = j \vee L^i_\mu \neq \neg L^j_\nu) \tag{6.2}$$

Solution grids in this compatibility matrix are in one-to-one relation with the index-distinct implicants in the following DF of formula 6.1:

$$f = \bigvee_{(\mu_1,\mu_2,\ldots,\mu_n)\in[m_1]\times[m_2]\times\ldots\times[m_n]} L^1_{\mu_1} \wedge L^2_{\mu_2} \wedge \ldots \wedge L^n_{\mu_n}$$

The one-to-one relation is:

$$\phi \overset{1-1}{\longleftrightarrow} L^1_{\phi(1)} \wedge L^2_{\phi(2)} \wedge \ldots \wedge L^n_{\phi(n)}$$

- where function ϕ is the index function of a solution grid.

For SAT, Theorem 3.1 becomes exactly the Cook's 3SAT theorem. Then, Theorem 2.7 positively resolves the P vs NP problem and more:

$$\mathrm{P} = \mathrm{NP} \subseteq \mathrm{P/poly}$$

And, in full compliance with Fortune and Mahaney's theorems [15, 16], monotone circuits 2.12 and the congruence classes of linear systems 4.4 are examples of the NP-complete sparse set.

Exercise 6.1. There is not CNF whose compatibility matrix 6.2 is compatibility matrix from Exercise 1.3.

Really, let's assume the opposite. Let f be a CNF whose compatibility matrix 6.2 is the above matrix B. Then,

$$f = (L_1^1 \vee L_2^1) \wedge (L_1^2 \vee L_2^2) \wedge (L_1^3 \vee L_2^3)$$

Then, there is the following contradiction:

$$\left.\begin{array}{lll} b_{12}^{12} = \mathit{false} & \Rightarrow & L_1^1 = \neg L_2^2 \\ b_{11}^{13} = \mathit{false} & \Rightarrow & L_1^1 = \neg L_1^3 \\ b_{21}^{23} = \mathit{false} & \Rightarrow & L_2^2 = \neg L_1^3 \end{array}\right\} \Rightarrow \neg L_2^2 = L_1^1 = \neg L_1^3 = L_2^2$$

6.2. **Subgraph isomorphism.** Let $g_1 = (V_1, E_1)$ and $g_2 = (V_2, E_2)$ be two graphs. The problems is to decide whether or not there is injection $\psi : V_1 \to V_2$ such that

$$(u_1, u_2) \in E_1 \Rightarrow (\psi(u_1), \psi(u_2)) \in E_2$$

Let's arbitrarily enumerate the graphs' vertex sets:

$$V_1 = \{u_1, u_2, \dots, u_{|V_1|}\},\ V_2 = \{w_1, w_2, \dots, w_{|V_2|}\}$$

Let $A_1 = (a_{ij}^1)_{|V_1| \times |V_1|}$ and $A_2 = (a_{ij}^2)_{|V_2| \times |V_2|}$ be the appropriate *Boolean adjacency matrices* of graphs g_1 and g_2:

$$a_{ij}^1 = ((u_i, u_j) \in E_1),\ a_{ij}^2 = ((v_i, v_j) \in E_2)$$

The following box matrix $B = (B_{ij})_{|V_1| \times |V_1|}$ is a compatibility matrix for the Subgraph isomorphism problem:

$$B_{ij} = \begin{cases} I_{|V_2|}, & i = j \\ \neg I_{|V_2|}, & i \neq j \wedge \neg a_{ij}^1 \\ A_2, & i \neq j \wedge a_{ij}^1 \end{cases} \tag{6.3}$$

- where $I_{|V_2|}$ is the $|V_2| \times |V_2|$ Boolean identity matrix[15], and matrix $\neg I_{|V_2|}$ is the per-component negation of matrix $I_{|V_2|}$. Solution grids in this compatibility matrix are in one-to-one relation with the injections in question:

$$\phi \xleftrightarrow{1-1} \psi = (u_1 \mapsto w_{\phi(1)}, u_2 \mapsto w_{\phi(2)}, \dots, u_{|V_1|} \mapsto w_{\phi(|V_1|)})$$

- where function ϕ is the index function of a solution grid.

[15] All diagonal components equal *true*, and all off-diagonal components equal *false*.

Exercise 6.2 (Clique). For the k-Clique problem adjacency matrix of graph g_1 is

$$A_1 = \begin{pmatrix} 0 & 1 & 1 & \ddots & 1 & 1 & 1 \\ 1 & 0 & 1 & \ddots & 1 & 1 & 1 \\ 1 & 1 & 0 & \ddots & 1 & 1 & 1 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 1 & 1 & 1 & \ddots & 0 & 1 & 1 \\ 1 & 1 & 1 & \ddots & 1 & 0 & 1 \\ 1 & 1 & 1 & \ddots & 1 & 1 & 0 \end{pmatrix}_{k\times k}$$

And compatibility matrix 6.3 is

$$B = \begin{pmatrix} I_{|V_2|} & A_2 & A_2 & \ddots & A_2 & A_2 & A_2 \\ A_2 & I_{|V_2|} & A_2 & \ddots & A_2 & A_2 & A_2 \\ A_2 & A_2 & I_{|V_2|} & \ddots & A_2 & A_2 & A_2 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ A_2 & A_2 & A_2 & \ddots & I_{|V_2|} & A_2 & A_2 \\ A_2 & A_2 & A_2 & \ddots & A_2 & I_{|V_2|} & A_2 \\ A_2 & A_2 & A_2 & \ddots & A_2 & A_2 & I_{|V_2|} \end{pmatrix}_{k\times k}$$

This compatibility-matrix encoding is Karp reduction of the k-Clique problem to the k-Clique problem in a k-partite graph. And this k-partiteness unfastens Razborov's super-polynomial bound [17] on the problem's circuit complexity (later tightened by Alon and Boppana [18]) with the $O(k^2n^2)$ ”slack variables” added to Razborov's indicator f of the C_n^2 arguments.

The solution grids problem itself is an instance of the n-Clique problem in a n-partite graph with $O(mn)$ vertices: we just nil in compatibility matrix 1.1 all its diagonal components to obtain an adjacency matrix of that graph. The n-partiteness of this model is the major benefit of the compatibility-matrix encoding (See Exercise 2.1).

Exercise 6.3 (Graph isomorphism). Graph isomorphism problem is the Subgraph isomorphism instance when $|V_1| = |V_2|$ and $|E_1| = |E_2|$.

Exercise 6.4 (Path and Cycle). The k-Path/Cycle problem is a Subgraph isomorphism instance when graph g_1 has adjacency matrix

$$A_1 = \begin{pmatrix} 0 & 1 & 0 & \ddots & 0 & 0 & c \\ 1 & 0 & 1 & \ddots & 0 & 0 & 0 \\ 0 & 1 & 0 & \ddots & 0 & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \ddots & 0 & 1 & 0 \\ 0 & 0 & 0 & \ddots & 1 & 0 & 1 \\ c & 0 & 0 & \ddots & 0 & 1 & 0 \end{pmatrix}_{k \times k}$$

- where $c = 0$ for the Path and $c = 1$ for the Cycle. Compatibility matrix 6.3 is

$$B = \begin{pmatrix} I_{|V_2|} & A_2 & \neg I_{|V_2|} & \ddots & \neg I_{|V_2|} & \neg I_{|V_2|} & C \\ A_2 & I_{|V_2|} & A_2 & \ddots & \neg I_{|V_2|} & \neg I_{|V_2|} & \neg I_{|V_2|} \\ \neg I_{|V_2|} & A_2 & I_{|V_2|} & \ddots & \neg I_{|V_2|} & \neg I_{|V_2|} & \neg I_{|V_2|} \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \neg I_{|V_2|} & \neg I_{|V_2|} & \neg I_{|V_2|} & \ddots & I_{|V_2|} & A_2 & \neg I_{|V_2|} \\ \neg I_{|V_2|} & \neg I_{|V_2|} & \neg I_{|V_2|} & \ddots & A_2 & I_{|V_2|} & A_2 \\ C & \neg I_{|V_2|} & \neg I_{|V_2|} & \ddots & \neg I_{|V_2|} & A_2 & I_{|V_2|} \end{pmatrix}_{k \times k}$$

- where $C = \neg I_{|V_2|}$ for the Path and $C = A_2$ for the Cycle.

The Hamiltonian path/cycle problem is the instance when $k = |V_2|$.

6.3. **QSAT.** Let F be the following quantified CNF in prenex normal form:

$$F = Q_1 x_1 Q_2 x_2 \ldots Q_k x_k f \tag{6.4}$$

- where matrix f is CNF 6.1; $Q_1, Q_2, \ldots, Q_k \in \{\forall, \exists\}$ are quantifiers; the number of quantifiers equal the number of variables in formula f, $k = |A|$; and variables $x_1, x_2, \ldots, x_k$ in formula f are enumerated in the order of the quantification. The problem is to compute F.

The following CF is equivalent CNF f:

$$f' = f \ \wedge \ \bigwedge_{l=1}^{k} (x_l \vee \bar{x}_l) \ \Leftrightarrow \ f$$

Let's compute truth table for each clause c_i in CF f', $i \in [n+k]$:

$T_i =$

#	True assignments	Clause c_i
1	The 1st true assignment to the variables in clause c_i	The 1st meaning of clause c_i
$\vdots$	$\vdots$	$\vdots$
$\lvert T_i \rvert$	The $\lvert T_i \rvert$th true assignment to the variables in clause c_i	The $\lvert T_i \rvert$th meaning of clause c_i

- where length $|T_i|$ of table T_i equals $2^{|c_i|}$ for the clause c_i in CNF f ($|c_i|$ is the length of clause c_i and $i \in [n]$), and $|T_i| = 2$ for the added clauses $x_l \vee \bar{x}_l$ ($l \in [k]$, $i = n+1, n+2, \ldots, n+k$). In the terms of system 4.1, the numbers of strings in table T_i will be options for parts in the solution partition aka clause c_i. And the appropriate compatibility relations are: the μth string in truth table T_i and the νth string in truth table T_j are compatible if meanings of all variables in these true assignments do not contradict each other (the same variables have the same true assignment), and the μth and $\nu$the meanings of clauses c_i and c_j both are *true*. Let's graph all these compatibility relations in the appropriate spaces $[|T_i|] \times [|T_j|]$ and aggregate those graphics in a box matrix in accordance with the clauses' indices:

$$B = (B_{ij})_{n\times n} = ((b^{ij}_{\mu\nu})_{|T_i|\times|T_i|})_{(n+k)\times(n+k)} \tag{6.5}$$

Compatibility boxes in compatibility matrix B are arranged as follows:

$$B = \left(\begin{array}{c|c} \begin{array}{c}\text{Compatibility boxes}\\ \text{for clauses in CNF } f\\ \text{in the clauses' order}\end{array} & \begin{array}{c}\text{Mixed}\\ \text{compatibility}\\ \text{boxes}\end{array} \\ \hline \begin{array}{c}\text{Mixed}\\ \text{compatibility}\\ \text{boxes}\end{array} & \begin{array}{c}\text{Compatibility boxes}\\ \text{for clauses } x_l \vee \bar{x}_l,\\ l = 1, 2, \ldots, k\end{array} \end{array} \right)_{(n+k)\times(n+k)}$$

Solution grids in compatibility matrix B are in one-to-one relation with the true assignments satisfying both formulae f' and f:

$$\phi \overset{1-1}{\longleftrightarrow} \left\{ \begin{array}{l} \text{variables in clause } c_i \text{ have} \\ \text{their } \phi(i)\text{th true assignment from} \\ \text{the truth table for clause } c_i \end{array} \right.$$

- where ϕ is index function of a solution grid in compatibility matrix B and, thus, the same variables in different clauses have the same assignments.

QCNF 6.4 can be expressed by SAT instance 3.2 for compatibility matrix 6.5 subject to the following constrains on compatibility boxes B_{ij}, $i, j > n$ (these 2×2 compatibility boxes are located in the lower-right corner, and these constrains express leaves in the splitting tree of formula F):

$i<j$	$Q_{j-n}=\forall$	$Q_{j-n}=\exists$
$Q_{i-n}=\forall$	$b^{ij}_{11}\wedge b^{ij}_{12}\wedge b^{ij}_{21}\wedge b^{ij}_{22}=true$	$(b^{ij}_{11}\vee b^{ij}_{12})\wedge(b^{ij}_{21}\vee b^{ij}_{22})=true$
$Q_{i-n}=\exists$	$b^{ij}_{11}\wedge b^{ij}_{12}\vee b^{ij}_{21}\wedge b^{ij}_{22}=true$	$b^{ij}_{11}\vee b^{ij}_{12}\vee b^{ij}_{21}\vee b^{ij}_{22}=true$

The following substitutions may be done in these constrains:

$$b^{ij}_{\mu\nu}=\chi^i_\mu\wedge\chi^j_\nu$$

- where χ^i_μ and χ^j_ν are the appropriate variables in formula 3.2 for compatibility matrix 6.5. And the constrains may be replaced by their CNF-expressions and added to formula 3.2. There will be $O(m^2n^2)$ clauses in the resulting CNF, where $O(mn)$ is the size of matrix f. And satisfiability of that CNF will indicate whether or not $F=true$. Therefore,

$$\text{PSPACE}=\text{NP}$$

- compatibility matrix method Karp reduced QSAT to SAT.

6.4. **Reachability.** Let $g=(V,E)$ be a (multi)digraph (loops are allowed). The problem is to decide whether or not there are k-walks and k-paths starting in $V_1\subseteq V$ and finishing in $V_2\subseteq V$.

Let $A=(a_{\mu\nu})_{|V|\times|V|}$ be a Boolean adjacency matrix of digraph g (see Subsection 6.2). Let's define powers[16] of matrix A:

$$A^0=I_{|V|};\ A^p=A^{p-1}A,\ p>0;\ A^p=(A^{-p})^T,\ p<0$$

- where $I_{|V|}$ is the $|V|\times|V|$ Boolean identity matrix. Let's D_1 and D_2 be Boolean diagonal matrices indicating sets V_1 and V_2 appropriately. Then, the following box matrix is a compatibility matrix for the problem:

$$B=(B_{ij})_{(k+1)\times(k+1)}$$

- where

$$B_{ij}=\begin{cases} D_1, & i=j=1\\ I_{|V|}, & i=j\notin\{1,k+1\}\\ D_2, & i=j=k+1\\ A^{j-i}, & i\neq j \text{ and it is the Walk problem}\\ \neg I_{|V|}\wedge A^{j-i}, & i\neq j \text{ and it is the Path problem}\end{cases}\tag{6.6}$$

Solution grids in this compatibility matrix are in one-to-one relation with the k-walks/paths in digraph g:

$$\phi\overset{1-1}{\longleftrightarrow}(v_{\phi(1)},v_{\phi(2)},\ldots,v_{\phi(k)},v_{\phi(k+1)})\tag{6.7}$$

- where function ϕ is the index function of a solution grid. This example will help us to find domain of the compatibility matrix method and more.

[16] We use the following Boolean matrix multiplication: $(a_{i\mu})(b_{\mu j})=(\bigvee_\mu a_{i\mu}\wedge b_{\mu j})$.

Let $M = (\Sigma, \Gamma, Q, \delta, Q_{\text{init}}, Q_{\text{accept}}, Q_{\text{reject}})$ be a Turing machine, deterministic or not:

$$\delta \subseteq (Q - (Q_{\text{accept}} \cup Q_{\text{reject}})) \times \Gamma \times Q \times \Gamma \times \{\leftarrow, \rightarrow\}$$

Let function $t_M : s \in \Sigma^* \mapsto t_M(s) \in \mathbf{N} \cup \{\infty\} = \{0, 1, 2, \ldots, \infty\}$ be the time during which machine M computes strings $s \in \Sigma^*$ and accept or reject them. Let T_M be the time-complexity of machine M:

$$T_M = \sup\{t_M(s) \mid s \in \Sigma^*\}$$

Let's assume that machine M is such that

$$T_M < \infty$$

Then, we can compute compatibility matrix 6.6 of the T_M-walks for the following digraph g and sets V_1 and V_2:

$$g = (Q, Q \times Q),\ V_1 = Q_{\text{init}},\ V_2 = Q_{\text{accept}}$$

Let B be that compatibility matrix:

$$B = ((b^{ij}_{\mu\nu})_{|Q| \times |Q|})_{(T_M+1)\times(T_M+1)}$$

Solution grids in compatibility matrix B are in one-to-one relation 6.7 with the T_M-walks in digraph g. Let $C = \{c\}$ be the set of all computational threads of machine M. Then, $|c| \leq T_M + 1$ for any $c \in C$. And each $c \in C$ creates a $(|c| - 1)$-walk in digraph g. Let's agree to continue each such walk by the $T_M - |c| + 1$ loops in the case of $|c| < T_M + 1$. And let w_c be that T_M-walk in digraph g (possibly ending with the added loops) which was created by computational thread $c \in C$. Then, there are index functions ϕ_c in compatibility matrix B which are matched to walks w_c by one-to-one relation 6.7. Then, the following compatibility matrix does exist:

$$B' = ((0)_{|Q|\times|Q|})_{(T_M+1)\times(T_M+1)} \vee \bigvee_{c \in C} ((b^{ij}_{\phi_c(i)\phi_c(j)})_{|Q|\times|Q|})_{(T_M+1)\times(T_M+1)}$$

- where $b^{ij}_{\phi_c(i)\phi_c(j)}$ are the appropriate components of our compatibility matrix B. And solution grids in compatibility matrix B' are in one-to-one relation with the accepting computational threads of machine M:

$$\phi_c \overset{1-1}{\longleftrightarrow} w_c \overset{1-1}{\longleftrightarrow} c$$

- where c are accepting computational threads.

Theorem 6.5. *The compatibility matrix method's domain consists of those Turing-decidable problems whose time-complexity is finite. And any such problem allows an adequate encoding by a compatibility matrix whose size is linear in the problem's time-complexity.*

Proof. We just have seen that any Turing-decidable problem of the finite time-complexity allows such an encoding. And on the other hand, we have seen in the previous sections, any problem which allows the compatibility-matrix encoding can be solved by a decider (Sipser's decider [19], Kozen's total Turing machine [20]) in the time polynomial in the compatibility matrix's size. □

Exercise 6.6. For the given NDTM of the finite time-complexity, we can replace the polynomial time-bound in the Garey and Johnson's proof [21] of Cook-Levin theorem [9, 22] by the machine's time-complexity. Then, the argument will produce a SAT instance with a quadratic number of clauses and variables, quadratic in the machine's time-complexity. And compatibility matrix 6.2 for that SAT instance will be another compatibility-matrix encoding of the given NDTM.

Theorem 6.5 validates all the above results for the Turing-decidable problems whose time-complexity is bounded by a finite function. And the results may be summarized as the following complements to Savitch's theorem [23]:

$$\mathrm{NTIME}(t) \subseteq \mathrm{DTIME}(t^8),\ \mathrm{NSPACE}(s) \subseteq \mathrm{DTIME}(s^{16})$$

And it directly proves

$$\begin{gathered} \mathrm{P} = \mathrm{NP} = \mathrm{PSPACE} \\ \mathrm{EXPTIME} = \mathrm{NEXPTIME} = \mathrm{EXPSPACE} \\ k\text{-}\mathrm{EXPTIME} = k\text{-}\mathrm{NEXPTIME} = k\text{-}\mathrm{EXPSPACE}, \quad k \geq 2 \end{gathered}$$

- and so on for other complexity classes closed under Karp reduction.

Compatibility-matrix encoding 1.1 is Post reduction of the encoded problem to the following finite recursive language[17]:

$$L = [m_1] \circ [m_2] \circ \ldots \circ [m_n]$$

And any finite recursive language allows the compatibility-matrix encoding due to Theorem 6.5. So, domain of the compatibility matrix method may be called FINITE.

7. Compatibility tensor

Turing-decidable problems can be encoded with their multidimensional views as well as with their 2D-views - it is often even easier. The multidimensional views will create the multidimensional Boolean box arrays aka *Boolean tensors.* And these blueprints can be formalized as follows.

[17] "o" is the concatenation operator.

Definition 7.1. *Compatibility tensor* T of the $\binom{k}{k}$-valency is a symmetric k-dimensional array of the Boolean k-dimensional arrays:

$$
(7.1) \qquad \begin{cases}
1). & T & = & (T_{i_1 i_2 \dots i_k})_{\underbrace{n \times n \times \dots \times n}_{k}} \\
2). & T_{i_1 i_2 \dots i_k} & = & (t^{i_1 i_2 \dots i_k}_{\mu_1 \mu_2 \dots \mu_k})_{m_1 \times m_2 \times \dots m_k} \\
3). & t^{i_1 i_2 \dots i_k}_{\mu_1 \mu_2 \dots \mu_k} & \in & \{false, true\} \\
4). & t^{\dots i \dots j \dots}_{\dots \mu \dots \nu \dots} & = & t^{\dots j \dots i \dots}_{\dots \nu \dots \mu \dots} \\
5). & t^{i_1 i_2 \dots i_k}_{\mu_1 \mu_2 \dots \mu_k} & \Rightarrow & \bigwedge_{\alpha,\beta \in [k]} (i_\alpha \neq i_\beta \ \vee \ \mu_\alpha = \mu_\beta)
\end{cases}
$$

- where numbers $n, m_1, m_2, \dots, m_n$ and maximum $m = \max_{i \in [n]} m_i$ are the tensor's sizes; and number k is the tensor's dimension. The k-dimensional boxes[18] $T_{i_1 i_2 \dots i_k}$ are called compatibility boxes. The components $t^{i_1 i_2 \dots i_k}_{\mu_1 \mu_2 \dots \mu_k}$ which equals true are *true*-components, and the rest components of the tensor are *false*-components.

The tensor's index function is any one-meaning function

$$\phi : \ i \in [n] \ \mapsto \ \phi(i) \in [m_i]$$

Each index function creates the following k-dimensional sub-array of the tensor which is called grid of components or just grid:

$$\{t^{i_1 i_2 \dots i_k}_{\phi(i_1)\phi(i_2)\dots\phi(i_k)}\}_{i_1, i_2, \dots, i_k \in [n]}$$

This grid of components is a solution grid if all its components equals *true*, i.e. if its index function satisfies the following functional equation:

$$(7.2) \qquad t^{i_1 i_2 \dots i_k}_{\phi(i_1)\phi(i_2)\dots\phi(i_k)} = true, \ i_1, i_2, \dots, i_k \in [n]$$

- solution grid in compatibility tensor T is a k-dimensional orthogonal lattice of *true*-components, one component per compatibility box.

Further, *true*-component $t^{i_1^0 i_2^0 \dots i_k^0}_{\mu_1^0 \mu_2^0 \dots \mu_k^0} = true$ is a noise if functional equation 7.2 is inconsistent subject to the following constrains

$$\phi(i_1^0) = \mu_1^0, \ \phi(i_2^0) = \mu_2^0, \ \dots, \ \phi(i_k^0) = \mu_k^0$$

Otherwise, *true*-component $t^{i_1^0 i_2^0 \dots i_k^0}_{\mu_1^0 \mu_2^0 \dots \mu_k^0}$ belongs to a solution grid.

Inversion of all noisy components in compatibility tensor will transform it into its general solution. The general solution is a compatibility tensor on its own. And the general solution of tensor 7.1 contains *true*-components iff compatibility tensor 7.1 contains solution grids.

Exercise 7.2. Any compatibility matrix 1.1 is a compatibility tensor of the $\binom{2}{2}$-valency; the matrix's major diagonal is a compatibility tensor of

[18] Some of the boxes will dimensionally degenerate when $m_i = 1$ for some $i \in [n]$.

the $\binom{1}{1}$-valency; and for any $k \geq 1$, the following conjunctions create a compatibility tensor of the $\binom{k}{k}$-valency:

$$t^{i_1 i_2 \ldots i_k}_{\mu_1 \mu_2 \ldots \mu_k} = \bigwedge_{(i,\mu),(j,\nu) \in \{(i_1,\mu_1),(i_2,\mu_2),\ldots,(i_k,\mu_k)\}} b^{ij}_{\mu\nu}$$

- where $i_1, i_2, \ldots, i_k \in [n]$ and $(\mu_1, \mu_2, \ldots, \mu_k) \in [m_{i_1}] \times [m_{i_2}] \times \ldots \times [m_{i_k}]$.

Compatibility tensors are used for encoding of problems exactly as the compatibility matrices. And the only requirement is the encoding adequacy: there are solution grids in the *compatibility-tensor encoding* of the given problem iff that problem is consistent, i.e. iff it has solutions.

Exercise 7.3. Let compatibility relations in system 4.1 be k-ary relations:

$$\rho_{i_1 i_2 \ldots i_k} \subseteq O_{i_1} \times O_{i_2} \times \ldots \times O_{i_k}$$

- where (partially) diagonal relations are the appropriate (partially) inclusion relations. Then, the k-ary graphics of these relations will create an adequate compatibility-tensor encoding of the system.

Everything from the previous sections can be edited for the compatibility tensors. Yet, let's notice, the solution grids existence problem for compatibility tensor 7.1 is Turing-decidable: we just iterate all $\prod_{i \in [n]} m_i$ index functions of the tensor and test them with equation 7.2. Therefore, due to Theorem 6.5, any compatibility tensor allows the compatibility-matrix encoding. The encoding can be done with the following protocol: *in the compatibility matrix of a compatibility tensor, the matrix's off-diagonal compatibility boxes encode the tensor's box structure while the matrix's diagonal compatibility boxes encode the tensor's components.*

Exercise 7.4. For compatibility tensor 7.1, the protocol can be realized as follows (in the enumeration invariant form):

$$B = (B_{\alpha_1 \alpha_2})_{n^k \times n^k} = ((b^{\alpha_1 \alpha_2}_{\beta_1 \beta_2})_{\prod_{\omega=1}^k m_{i^1_\omega} \times \prod_{\omega=1}^k m_{i^2_\omega}})_{n^k \times n^k} \tag{7.3}$$

- where

$$\begin{aligned}
\alpha_1 &= (i^1_1, i^1_2, \ldots, i^1_k) \in [n]^k; \\
\alpha_2 &= (i^2_1, i^2_2, \ldots, i^2_k) \in [n]^k; \\
\beta_1 &= (\mu^1_1, \mu^1_2, \ldots, \mu^1_k) \in [m_{i^1_1}] \times [m_{i^1_2}] \times \ldots \times [m_{i^1_k}]; \\
\beta_2 &= (\mu^2_1, \mu^2_2, \ldots, \mu^2_k) \in [m_{i^2_1}] \times [m_{i^2_2}] \times \ldots \times [m_{i^2_k}];
\end{aligned}$$

and

$$b^{\alpha_1 \alpha_2}_{\beta_1 \beta_2} = (\alpha_1 \neq \alpha_2 \vee t^{\alpha_1}_{\beta_1}) \wedge (\alpha_1 = \alpha_2 \vee \bigwedge_{\omega_1, \omega_2 \in [k]} (i^1_{\omega_1} \neq i^2_{\omega_2} \vee \mu_{i^1_{\omega_1}} = \mu_{i^2_{\omega_2}}));$$

where $t^{\alpha_1}_{\beta_1} = t^{i^1_1 i^1_2 \ldots i^1_k}_{\mu^1_1 \mu^1_2 \ldots \mu^1_k}$ are the tensor's components.

For example, compatibility matrix 2.4 is a rationalization of compatibility matrix 7.3 for compatibility tensor 1.1.

8. Conclusion

This article presented the compatibility matrix method a generalization of descriptive geometry on the combinatorial problems. In the method, problems are encoded by the "blueprints" of their solutions, and the solutions themselves become an orthogonal-lattice pattern in the encoding. Such a regularization of the problems allows perform parallel testing of all guesses with the efficient deterministic algorithms - the article presented a few. The encoding is Post reduction of Turing-decidable problems to finite recursive languages, and computational complexity of the compatibility matrix method is basically polynomial in the computational complexity of that reduction. Further, domain of the compatibility matrix method is exactly the class of finite recursive languages. The class was called FINITE. And, because any Rado's busy beaver [24] belongs to FINITE (it is easy to see),

$$\text{ELEMENTARY} \subset \text{FINITE} \subseteq \text{R}$$

And, the compatibility matrix method shows, the time and space complexities of the problems in class FINITE are polynomially equivalent.

There are some demos of the compatibility matrix method at [25].

References

[1] Sergey Gubin, *Complementary to Yannakakis' theorem,* JCMCC 74 (2010), pp. 313-321 (see arXiv:cs/0610042v3 [cs.DM])

[2] Sergey Gubin, A Sudoku solver, Proceedings of the World Congress on Engineering and Computer Science 2009 Vol I WCECS 2009, October 20-22, 2009, San Francisco, USA, ISBN: 978-98817012-6-8, pp. 223-228

[3] Sergey Gubin, *Polynomial size asymmetric linear model for SAT,* In Proc. of the Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008 (WCECS 2008), ISBN-13: 978-0-7695-3555-5, pp. 62-66 `http://doi.ieeecomputersociety.org/10.1109/WCECS.2008.16`

[4] Sergey Gubin, *Polynomial size asymmetric linear model for Subgraph Isomorphism,* Proceedings of the World Congress on Engineering and Computer Science 2009, WCECS 2008, ISBN: 978-988-98671-0-2, pp. 241-246 (see arXiv:0802.2612v2 [cs.DM])

[5] Sergey Gubin, *Compatibility matrix method,* 25-nd MCCCC, Program and Abstracts, p. 9, University of Nevada, Las Vegas, Nevada, 2011

[6] Sergey Gubin, *On the purely logical solution of Sudoku,* 24-nd MCCCC, Program and Abstracts, p. 9, Illinois State University, Normal, Illinois, 2010

[7] Sergey Gubin, *A lexicographical approach to SAT,* 24-nd MCCCC, Program and Abstracts, p. 9, Rochester Institute of Technology, Rochester, New York, 2008

[8] Stephen Cook, *The P versus NP Problem,* `http://www.claymath.org/millennium/P_vs_NP/Official_Problem_Description.pdf`

[9] Stephen Cook, The complexity of theorem-proving procedures. In Conference Record of Third Annual ACM Symposium on Theory of Computing, pages 151-158, 1971

[10] Richard M. Karp, *Reducibility among combinatorial problems.* In R. E. Miller and J. W. Thatcher, editors, Complexity of Computer Computations, pages 85103, New York, 1972. Plenum Press

[11] Schaefer, Thomas J. (1978), *The Complexity of Satisfiability Problems.* STOC 1978. pp. 216-226

[12] Mihalis Yannakakis, *Expressing combinatorial optimization problems by linear programs,* In Proc. of the twentieth annual ACM Sympos. on Theory of computing, Chicago, Illinois, pp. 223-228, 1988

[13] Grover L.K., *A fast quantum mechanical algorithm for database search,* Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (May 1996) p. 212

[14] T. Baker, J. Gill, and R. Solovay, *Relativizations of the P =? NP question.* SICOMP: SIAM Journal on Computing, 1975

[15] S. Fortune, *A note on sparse complete sets.* SIAM Journal on Computing, volume 8, issue 3, pp.431-433. 1979

[16] S. R. Mahaney, *Sparse complete sets for NP: Solution of a conjecture by Berman and Hartmanis.* Journal of Computer and System Sciences 25:130-143. 1982

[17] Alexander A. Razborov, *Lower bounds on the monotone complexity of some boolean functions.* Soviet Math. Dokl., 31:354-357, 1985

[18] N. Alon and R. B. Boppana, *The monotone circuit complexity of boolean functions.* Combinatorica, 7(1):1-22, 1987

[19] Sipser, M. (1996), *Introduction to the Theory of Computation,* PWS Publishing Co.

[20] Kozen, D.C. (1997), *Automata and Computability,* Springer

[21] M. R. Garey and D. S. Johnson, *Computers and Intractability, a Guide to the Theory of NP-Completeness.* W.H. Freeman and Co., San Francisco, 1979

[22] L. Levin. *Universal search problems (in Russian).* Problemy Peredachi Informatsii, 9(3):265-266, 1973. English translation in Trakhtenbrot, B. A.: *A survey of Russian approaches to Perebor (brute-force search) algorithms.* Annals of the History of Computing, 6 (1984), pages. 384-400

[23] Savitch, Walter J. (1970), *Relationships between nondeterministic and deterministic tape complexities,* Journal of Computer and System Sciences 4 (2):177-192

[24] Radó, Tibor (1962), On non-computable functions, Bell System Technical Journal, Vol. 41, No. 3 (May 1962), pp. 877-884

[25] Polynomial times, `http://www.timescube.com`