$1^{t-2}0^{k-t-1}110, 1^{t-2}0^{k-t-1}101$, and $1^{t-2}0^{k-t-1}011$

eventual output: all odd weight tuples

□

With Proposition 25 in hand, we can prove Theorems 13 and 14.

Theorem (Theorem 13 restated)**.** Let $k \geq 3$ and let $\mathbf{T} \subseteq \{0,1\}^k$ be a relation such that $\textbf{t-in-k} \to \mathbf{T}$ and $\mathrm{CSP}(\mathbf{T})$ is NP-hard. Then $\mathrm{PCSP}(\textbf{t-in-k}, \mathbf{T})$ is tractable if and only if $\mathbf{T} = \mathbf{NAE}$.

Proof. By Proposition 16, we can assume that $\mathbf{T}$ is symmetric. If $\mathbf{T} = \mathbf{NAE}$ then $\mathrm{PCSP}(\textbf{t-in-k}, \mathbf{T})$ is tractable by Proposition 8. Otherwise, we show that $\mathrm{Pol}(\textbf{t-in-k}, \mathbf{T})$ does not contain any of the tractable polymorphism families identified in the symmetric Boolean PCSP dichotomy (Theorem 5), and therefore $\mathrm{PCSP}(\textbf{t-in-k}, \mathbf{T})$ is NP-hard.

The families we need to rule out are constants, OR, AND, XOR, AT, and THR_q for $q \in \mathbb{Q}$, as well as their negations. We deal first with the non-negated families. Since $\mathrm{CSP}(\mathbf{T})$ is NP-hard, by Proposition 25, we have $0^k \notin \mathbf{T}$ and $1^k \notin \mathbf{T}$. Hence, $\mathrm{Pol}(\textbf{t-in-k}, \mathbf{T})$ does not contain constants.

Let C_k^t be the $k \times k$ matrix containing the k cyclic shifts of the column 1^t0^{k-t}. Then C_k^t prevents the polymorphism families OR, AND, XOR (if k is odd), and THR_q for all $q \neq \frac{t}{k}$. The case $q = \frac{t}{k}$ is ruled out by [20, Fact B.3], and AT is ruled out by [10, Claim 4.6]. Since $\mathrm{CSP}(\mathbf{T})$ is NP-hard, by Proposition 25, it remains to show that for even k, $\mathrm{Pol}(\textbf{t-in-k}, \mathbf{T})$ excludes XOR when t is even, and likewise when t is odd and $\mathbf{T}$ is missing a tuple of odd weight.

Let k and t be even. Applying XOR_k to the matrix C_k^t returns the tuple 0^k, so applying XOR_{k-1} to the first $k-1$ columns of C_k^t returns the last column $1^{t-1}0^{k-t}1$. We can "fill in" the 0's in the output by swapping 0/1 pairs of values in the input matrix. In particular, in the columns $k-1, k-3, \ldots, t+1$, we swap the entries in the pairs of rows $(k-1, k-2), (k-3, k-4), \ldots, (t+1, t)$, respectively. The resulting $k \times (k-1)$ matrix M then satisfies $\mathrm{XOR}_{k-1}(M) = 1^k$ and the arity $k-1$ is odd as required. An example with swapped values in bold is illustrated in Figure (5a).

$$\mathrm{XOR}_7 \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & \mathbf{1} & 0 & 0 \\ 0 & 1 & 1 & 1 & \mathbf{0} & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & \mathbf{1} \\ 0 & 0 & 0 & 1 & 1 & 1 & \mathbf{0} \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} = \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix}$$

(a) $t = 4$ and $k = 8$.

$$\mathrm{XOR}_5 \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{matrix}$$

(b) $t = 3$, $k = 6$, and $d = 5$.

Figure 5: XOR.

Now let k be even, t be odd, and suppose that $\mathbf{T}$ does not contain the tuple $\mathbf{x} = 1^d0^{k-d}$ of odd weight d. By Observation 20, we can assume without loss of generality that $t < d$. Then

XOR_d applied to the matrix C_d^t padded with $k-d$ rows of 0's returns $\mathbf{x}$. An illustration is given in Figure (5b). Therefore $\mathrm{XOR} \not\subseteq \mathrm{Pol}(\mathbf{t\text{-}in\text{-}k}, \mathbf{T})$.

Negations: Let F be a family of functions. We reduce the task of showing $\overline{F} \not\subseteq \mathrm{Pol}(\mathbf{t\text{-}in\text{-}k}, \mathbf{T})$ to the already completed task of showing $F \not\subseteq \mathrm{Pol}(\mathbf{t\text{-}in\text{-}k}, \mathbf{T})$. Let $\mathbf{x} \in \{0,1\}^k \setminus \mathbf{T}$, let $f \in F$ be a function of arity m, and let M be a $k \times m$ matrix of inputs to f whose columns are **t-in-k** tuples. We established $F \not\subseteq \mathrm{Pol}(\mathbf{t\text{-}in\text{-}k}, \mathbf{T})$ by finding f and M with $f(M) = \mathbf{x}$, and in the remaining cases we must find $\overline{f} \in \overline{F}$ and M such that $\overline{f}(M) = \mathbf{x}$. But since $\overline{f}(M) = \mathbf{x} \Leftrightarrow f(M) = \overline{\mathbf{x}}$, it suffices to find $f \in F$ such that $f(M) = \overline{\mathbf{x}}$, where $\overline{\mathbf{x}} = (1 - x_1, \ldots, 1 - x_k)$ if $\mathbf{x} = (x_1, \ldots, x_k)$.

The families $\overline{\mathrm{AND}}$, $\overline{\mathrm{OR}}$, $\overline{\mathrm{XOR}}$ (except when k is even and t is odd), and $\overline{\mathrm{THR}_q}$ for all $q \neq \frac{t}{k}$ are excluded from $\mathrm{Pol}(\mathbf{t\text{-}in\text{-}k}, \mathbf{T})$ in the same way as AND, OR, XOR, and THR_q with the same matrices serving as counterexamples. In detail, the matrix C_k^t, which contains k cyclic shifts of the column $1^t 0^{k-t}$, prevents the polymorphism families $\overline{\mathrm{AND}}$, $\overline{\mathrm{OR}}$, $\overline{\mathrm{XOR}}$ (if k is odd), and $\overline{\mathrm{THR}_q}$ (if $q \neq \frac{t}{k}$). The case $\overline{\mathrm{XOR}}$ with k even, t even is ruled out the same way as before, illustrated in Figure (5a).

To see that $\overline{\mathrm{AT}}$ and $\overline{\mathrm{THR}_{\frac{t}{k}}}$ (with $q = \frac{t}{k}$) are also excluded, let $\mathbf{x} \notin \mathbf{T}$ be a tuple of weight $d \neq t$. Then the tuple $\overline{\mathbf{x}}$ of weight $k - d$ can be returned by an AT function [10, Claim 4.6] and a $\mathrm{THR}_{\frac{t}{k}}$ function [20, Fact B.3]. If $k - d = t$, then the AT and $\mathrm{THR}_{\frac{t}{k}}$ functions of arity 1 output $\overline{\mathbf{x}}$ on input $\overline{\mathbf{x}}$.

Finally, when k is even, t is odd, and $\mathbf{T}$ does not contain the tuple $\mathbf{x}$ of odd weight d, the XOR argument above (illustrated in Figure 5b) applies since $\overline{\mathbf{x}}$ also has odd weight $k - d$. Again, if $k - d = t$, then the XOR function of arity 1 outputs $\overline{\mathbf{x}}$ on input $\overline{\mathbf{x}}$. ◻

Theorem (Theorem 14 restated)**.** Let $k \geq 3$ and $\emptyset \neq S \subseteq (\mathbf{t\text{-}in\text{-}k})^c \cap \mathbf{NAE}$. If t is odd, k is even, and S contains tuples of only even weight, then $\mathrm{PCSP}(\mathbf{t\text{-}in\text{-}k}, \mathbf{NAE} \setminus \mathbf{S})$ is tractable. Otherwise, $\mathrm{PCSP}(\mathbf{t\text{-}in\text{-}k}, \mathbf{NAE} \setminus \mathbf{S})$ is NP-hard.

Proof. The tractability in the first statement of the theorem is proved in Section 3. Otherwise, t is even, or k is odd, or S contains a tuple of odd weight. Take $\mathbf{T} = \mathbf{NAE} \setminus \mathbf{S}$. Observe that case (1) of Proposition 25 does not apply as neither 0^k nor 1^k is part of the template. Moreover, case (2) of Proposition 25 does not apply either: If t is odd and k is even then S contains a tuple of odd weight and hence $\mathbf{NAE} \setminus \mathbf{S}$ cannot have all odd weight tuples. Thus, by Proposition 25, $\mathrm{CSP}(\mathbf{T})$ is NP-hard. Then, by Theorem 13, $\mathrm{PCSP}(\mathbf{t\text{-}in\text{-}k}, \mathbf{T}) = \mathrm{PCSP}(\mathbf{t\text{-}in\text{-}k}, \mathbf{NAE} \setminus \mathbf{S})$ is NP-hard. ◻

Acknowledgements

We would like to thank the anonymous referees of both the conference [13] and this full version of the paper. We also thank Kristina Asimi and Libor Barto for useful discussions regarding the content of this and their paper [1].

References

[1] Kristina Asimi and Libor Barto. Finitely tractable promise constraint satisfaction problems. In *Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science (MFCS'21)*, volume 202 of *LIPIcs*, pages 11:1–11:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.MFCS.2021.11.

[2] Albert Atserias and Víctor Dalmau. Promise Constraint Satisfaction and Width. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*, pages 1129–1153, 2022. arXiv:2107.05886, doi:10.1137/1.9781611977073.48.

[3] Per Austrin, Amey Bhangale, and Aditya Potukuchi. Improved inapproximability of rainbow coloring. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 1479–1495, 2020. arXiv:1810.02784, doi:10.1137/1.9781611975994.90.

[4] Per Austrin, Venkatesan Guruswami, and Johan Håstad. $(2+\epsilon)$-Sat is NP-hard. *SIAM J. Comput.*, 46(5):1554–1573, 2017. doi:10.1137/15M1006507.

[5] Libor Barto, Diego Battistelli, and Kevin M. Berg. Symmetric Promise Constraint Satisfaction Problems: Beyond the Boolean Case. In *Proceedings of the 38th International Symposium on Theoretical Aspects of Computer Science (STACS'21)*, volume 187 of *LIPIcs*, pages 10:1–10:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. arXiv:2010.04623, doi:10.4230/LIPIcs.STACS.2021.10.

[6] Libor Barto, Jakub Bulín, Andrei A. Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. *Journal of the ACM*, 68(4):28:1–28:66, 2021. arXiv:1811.00970, doi:10.1145/3457606.

[7] Libor Barto, Andrei Krokhin, and Ross Willard. Polymorphisms, and how to use them. In Andrei Krokhin and Stanislav Živný, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2017. doi:10.4230/DFU.Vol7.15301.1.

[8] Libor Barto, Jakub Opršal, and Michael Pinsker. The wonderland of reflections. *Israel Journal of Mathematics*, 223(1):363–398, Feb 2018. arXiv:1510.04521, doi:10.1007/s11856-017-1621-9.

[9] Joshua Brakensiek and Venkatesan Guruswami. An Algorithmic Blend of LPs and Ring Equations for Promise CSPs. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*, pages 436–455. SIAM, 2019. arXiv:1807.05194, doi:10.1137/1.9781611975482.28.

[10] Joshua Brakensiek and Venkatesan Guruswami. Promise Constraint Satisfaction: Algebraic Structure and a Symmetric Boolean Dichotomy. *SIAM J. Comput.*, 50(6):1663–1700, 2021. doi:10.1137/19M128212X.

[11] Joshua Brakensiek, Venkatesan Guruswami, Marcin Wrochna, and Stanislav Živný. The power of the combined basic LP and affine relaxation for promise CSPs. *SIAM J. Comput.*, 49:1232–1248, 2020. arXiv:1907.04383, doi:10.1137/20M1312745.

[12] Alex Brandts, Marcin Wrochna, and Stanislav Živný. The complexity of promise SAT on non-Boolean domains. *ACM Trans. Comput. Theory*, 13(4):26:1–26:20, 2021. arXiv:1911.09065, doi:10.1145/3470867.

[13] Alex Brandts and Stanislav Živný. Beyond PCSP(1-in-3,NAE). In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP'21)*, volume 198 of *LIPIcs*, pages 121:1–121:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. arXiv:2104.12800, doi:10.4230/LIPIcs.ICALP.2021.121.

[14] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005. doi:10.1137/S0097539700376676.

[15] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, 2017. arXiv:1703.03021, doi:10.1109/FOCS.2017.37.

[16] Lorenzo Ciardo and Stanislav Živný. CLAP: A New Algorithm for Promise CSPs. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*, pages 1057–1068, 2022. arXiv:2107.05018, doi:10.1137/1.9781611977073.46.