Testing Expressibility is Hard

Ross Willard⋆

Pure Mathematics Department
University of Waterloo
Waterloo, Ontario N2L 3G1 Canada
http://www.math.uwaterloo.ca/~rdwillar

Abstract. We study the *expressibility problem*: given a finite constraint language Γ on a finite domain and another relation R, can Γ express R? We prove, by an explicit family of examples, that the standard witnesses to expressibility and inexpressibility (gadgets/formulas/conjunctive queries and polymorphisms respectively) may be required to be exponentially larger than the instances. We also show that the full expressibility problem is co-**NEXPTIME**-hard. Our proofs hinge on a novel interpretation of a tiling problem into the expressibility problem.

Key words: constraint, relation, expressive power, inverse satisfiability, structure identification, conjunctive query, primitive positive formula, polymorphism, domino system, nondeterministic exponential time

1 Introduction

Given a fixed set Γ of *basic* constraint relations for building constraint programs or satisfaction problems, there are typically other (perhaps useful) *implicit* relations which may treated as if they were actually present in Γ, without affecting the expressiveness or complexity of Γ.

For example, consider the toy constraint language $\Gamma = \{\to, U\}$ on the domain $D = \{0, 1, 2, 3, 4, 5\}$, where $\to$ is the binary relation pictured in Figure 1 and U is the unary relation $\{0, 3\}$.

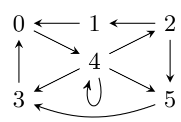

Fig. 1. The binary relation $\to$

⋆ The support of the Natural Sciences and Engineering Research Council of Canada and the American Institute of Mathematics is gratefully acknowledged.

The unary relation $V = \{3, 4, 5\}$ is an example of an implicit relation of $\{\to, U\}$. Indeed, whenever we wish to constrain a variable x to V, we can accomplish this by adding three new auxiliary variables a_x, b_x, c_x and imposing the basic constraints $a_x \to b_x$, $b_x \to x$, $x \to c_x$, $U(a_x)$, and $U(c_x)$. We say that Γ can *express* V. We might similarly ask: can Γ express the complement of V, i.e., the unary relation $W = \{0, 1, 2\}$? What about the complement of $\to$?

These questions are instances of the *expressibility problem* (also known as the *existential inverse satisfiability problem* [7, 6] and the *pp-definability problem* [4]). It is a *structure identification problem* in the sense of [8]. Its answers define what is called the *expressive power* of a constraint language [15], now a key tool in the quest to classify which constraint languages are tractable (e.g., [12, 3]).

In this paper we give constructions which show that the general expressibility problem is impossibly hard according to three natural measures.

We thank Matt Valeriote and Moshe Vardi for some helpful discussions.

2 Definitions, Basic Facts, and Statement of Results

Fix a constraint language Γ on a finite domain D. Given an instance $\mathcal{P} = (X, D, \mathcal{C})$ of CSP(Γ), we shall use Sol($\mathcal{P}$) to denote the set of all solutions to $\mathcal{P}$, construed as functions $X \to D$. If $\mathbf{s} = (s_1, \ldots, s_k)$ is a k-tuple of variables from X, then we shall use $\pi_{\mathbf{s}}(\mathrm{Sol}(\mathcal{P}))$ to denote the restriction of Sol($\mathcal{P}$) to $\mathbf{s}$; i.e.,

$$\pi_{\mathbf{s}}(\mathrm{Sol}(\mathcal{P})) = \{(f(s_1), \ldots, f(s_k)) \,:\, f \in \mathrm{Sol}(\mathcal{P})\} \;\subseteq D^k.$$

Definition 1 ([15, 5, 13]). *Given a constraint language Γ and a k-ary relation R on a domain D, we say that Γ* expresses *(or* generates*) R if there exists an instance $\mathcal{P} = (X, D, \mathcal{C})$ of* CSP(Γ) *and a k-tuple $\mathbf{s} = (s_1, \ldots, s_k)$ of variables with $\pi_{\mathbf{s}}(\mathrm{Sol}(\mathcal{P})) = R$. The pair $(\mathcal{P}, \mathbf{s})$ is a* witness *to the expressibility of R by Γ.*

Cohen and Jeavons [5] have called $\mathcal{P}$ a *gadget* and $\mathbf{s}$ a *construction site* in this context. A witness $(\mathcal{P}, \mathbf{s})$ can be trivially re-formulated as a *conjunctive query* over Γ (in database theory) or as a *primitive positive formula* over Γ (in logic); the latter is an expression of the form $\exists y_1 \cdots \exists y_n [C_1 \;\&\; C_2 \;\&\; \cdots \;\&\; C_r]$, asserting the existence of auxiliary variables satisfying (with $\mathbf{s}$) the constraints of P.

Example 1. In the example from Section 1, let $\mathcal{P}$ be the instance of CSP(Γ) having variable set $\{a, b, c, x\}$ and constraints $((a, b), \to)$, $((b, x), \to)$, $((x, c), \to)$, (a, U), and (c, U). $\mathcal{P}$ has exactly four solutions; identifying each solution f_i with its 4-tuple of values $(f_i(a), f_i(b), f_i(x), f_i(c))$, we have

$$\mathrm{Sol}(\mathcal{P}) = \{(0, 4, 3, 0),\ (0, 4, 4, 3),\ (0, 4, 5, 3),\ (3, 0, 4, 3)\}.$$

As the projection of Sol($\mathcal{P}$) on its third coordinate (i.e., at x) is $\{3, 4, 5\} = V$, $(\mathcal{P}, x)$ witnesses the fact that Γ can express V. An equivalent primitive positive formula witnessing this is $\exists a \exists b \exists c [a \to b \;\&\; b \to x \;\&\; x \to c \;\&\; U(a) \;\&\; U(c)]$.

Definition 2. *Suppose D is a finite domain, Γ is a constraint language over D, and n, k are positive integers. Let $\mathbf{s} = (\mathbf{s}_1, \ldots, \mathbf{s}_k)$ be a k-tuple of elements from D^n, R a k-ary relation on D, and $h : D^n \to D$.*

1. proj($\mathbf{s}$) $= \{(\mathbf{s}_1[i], \ldots, \mathbf{s}_k[i]) : 1 \leq i \leq n\}$. *(Thus* proj($\mathbf{s}$) $\subseteq D^k$.*)*
2. *h* preserves *R* at *s if* proj($\mathbf{s}$) $\not\subseteq R$ *or* $(h(\mathbf{s}_1), \ldots, h(\mathbf{s}_k)) \in R$.
3. *h* preserves *R if h preserves R at every k-tuple in $(D^n)^k$.*
4. *h is a* polymorphism *of Γ (of* arity *n) if h preserves every relation in Γ.*

One can show that, for every $n \geq 1$, there exists an instance of CSP(Γ) with variable set D^n whose solutions are precisely the n-ary polymorphisms of Γ. Following Jeavons, Cohen and Gyssens [14, 11, 15, 5], we call this CSP instance the *indicator problem for Γ of order n* and denote it by $\mathcal{I}_n(\Gamma)$.

It is well-known that the polymorphisms of Γ (i) include the projections and (ii) preserve all relations expressed by Γ (see e.g. [15, Lemma 2.18]). From this one can deduce the following connection between expressible relations, polymorphisms, and indicator problems.

Proposition 1. *For any $n, k \geq 1$ and $\mathbf{s} \in (D^n)^k$, the relation S expressed by $(\mathcal{I}_n(\Gamma), \mathbf{s})$ (i) contains* proj($\mathbf{s}$)*, and (ii) is contained in every k-ary relation expressible from Γ which contains* proj($\mathbf{s}$)*. I.e., S is the smallest k-ary relation expressible from Γ containing* proj($\mathbf{s}$)*.*

Note that if R is k-ary and there exists an n-ary polymorphism h of Γ which does not preserve R at some $\mathbf{s} \in (D^n)^k$, then R is not expressible from Γ. When this happens we say that h is a *witness* to the inexpressibility of R from Γ.

Example 2. Returning to the example in Section 1, the 1-ary map $h : D \to D$ when sends $1 \mapsto 3$, $2 \mapsto 4$, and fixes all other elements of D, is a polymorphism of $\Gamma = \{\rightarrow, U\}$. As $1 \in W = \{0, 1, 2\}$ but $h(1) \notin W$, h does not preserve W at 1; hence W is not expressible from Γ, and h is a witness.

For any k-ary relation R on D, if n is the number of rows of R (i.e., $n = |R|$), then one can construct $\mathbf{s}^{(R)} = (\mathbf{s}_1, \ldots, \mathbf{s}_k) \in (D^n)^k$ so that proj($\mathbf{s}^{(R)}$) $= R$. As R is expressible from Γ exactly when the smallest k-ary relation expressible from Γ and containing R is R itself, it follows from Proposition 1 that *either* $(\mathcal{I}_n(\Gamma), \mathbf{s}^{(R)})$ expresses R, *or* there exists an n-ary polymorphism of Γ which does not preserve R at $\mathbf{s}^{(R)}$. Thus we get the following theoretical upper bounds to the size of a witness to the expressibility or inexpressibility of R from Γ.

Corollary 1 ([9, 1, 15]). *Let $\Gamma \cup \{R\}$ be a set of relations on D, and let $n = |R|$.*

1. *If R is expressible from Γ, then R can be expressed by a* CSP *instance (or a primitive positive formula) with variable set of size $\leq |D|^n$.*
2. *R is* not *expressible from Γ if and only if there exists a polymorphism of Γ of arity $\leq n$ which does not preserve R.*

Example 3. Consider again the example in Section 1. The relation $V = \{3,4,5\}$ on the 6-element domain $\{0,1,2,3,4,5\}$ is expressible from $\Gamma = \{\to, U\}$, so Corollary 1 promises a CSP witness having $\leq 6^3 = 216$ variables. Conversely, the complement $\not\to$ of $\to$ turns out to be not expressible from Γ. Since $\not\to$ has 26 rows, Corollary 1 promises a witnessing polymorphism of arity ≤ 26.

Note the ridiculousness of the bounds in Example 3. Corollary 1 guarantees a CSP instance having ≤ 216 variables to express V, when in fact we have an instance using just 4 variables. Even worse is the promise of a 26-ary polymorphism witnessing the inexpressibility of $\not\to$; just storing the values of a random 26-ary function on $\{0,1,2,3,4,5\}$ would require over 5×10^8 terabytes. Yet the 1-ary polymorphism of Example 2 fails to preserve $\not\to$ (e.g., at $(2,2)$) and so already witnesses its inexpressibility.

Example 3 illustrates the fact that the upper bounds to the sizes of witnesses guaranteed by Corollary 1 are exponential in the size of the test relation. It is natural to ask if these upper bounds can be improved. For example, Cohen and Jeavons [5, p. 313] pose as an open research question the identification of circumstances under which sub-exponential sized CSP instances can be found witnessing expressible relations. Our first theorem says "not always":

Theorem 1. *For infinitely many n there exist a constraint language Γ_n and a relation R_n, both on a 22-element domain, such that $|R_n| = n$, R_n is expressible from Γ_n, but every* CSP(Γ_n) *instance expressing R_n has at least $2^{n/3}$ variables.*

Dually, our next theorem shows that in general we cannot hope to detect inexpressibility with sub-exponential sized polymorphisms.

Theorem 2. *For infinitely many n there exist a constraint language Γ'_n and a relation R'_n, both on a 22-element domain, such that $|R'_n| = n$, R'_n is not expressible from Γ'_n, but every witnessing polymorphism has arity at least $n/3$.*

We formally define Expr to be the combinatorial decision problem which takes as input a triple (D, Γ, R) (where D is a finite domain, Γ is a finite constraint language on D, and R is another relation on D), and asks whether R is expressible from Γ. Expr has also been called $\exists$-InvSat (the *existential inverse satisfiability problem*) [7, 6] and the *pp-definability problem* [4].

Corollary 1 and the discussion preceding it give a general algorithm for testing $\neg$Expr: among all functions $h : D^n \to D$ where $n = |R|$, search for one which (i) is a polymorphism of Γ, and (ii) does not preserve R at $\mathbf{s}^{(R)}$. This naive algorithm puts Expr in co-**NEXPTIME**. Dalmau [7, p. 163] speculated that perhaps there exists a better, more sophisticated algorithm which would place Expr in a lower complexity class. Suggestively, Creignou *et al* [6] have proved that Expr restricted to the boolean domain is in **P**.

At a workshop at AIM in 2008, a working group led by M. Vardi contrarily conjectured that there is essentially no algorithm better than the naive one, in the sense that Expr restricted to 3-element domains is co-**NEXPTIME**-complete [4]. In our last theorem we very nearly confirm this conjecture:

Theorem 3. *There exists $d > 1$ such that* Expr *restricted to d-element domains is co-**NEXPTIME**-complete.*

The remainder of this paper is devoted to proving Theorems 1–3 via an interpretation of certain tiling problems defined by domino systems.

3 Domino Systems and Tiling Problems

A *tiling problem* is a particular kind of constraint satisfaction problem whose constraints are organized "horizontally and vertically." More precisely:

Definition 3 ([10, 2]). *A* domino system *is a triple $\mathcal{D} = (\Delta, H, V)$ where Δ is a finite nonempty set (of "tile types") and H, V are binary relations on Δ (called the* horizontal *and* vertical adjacency constraint relations*).*

Notation 4. For $N > 1$ we will use $[N{\times}N]$ to denote the set

$$[N{\times}N] = \{(i, j) \,:\, i, j \in \mathbb{Z},\ 0 \le i, j < N\}.$$

We informally identify the element $(i, j) \in [N{\times}N]$ with the unit square in the x-y plane whose lower-left corner has coordinates (i, j). The kth *row* of $[N{\times}N]$ is the subset $\mathrm{Row}_k = \{(i, k) \,:\, 0 \le i < N\}$, while the kth *column* is the subset $\mathrm{Col}_k = \{(k, j) \,:\, 0 \le j < N\}$. Figure 2 illustrates the board $[4{\times}4]$.

Definition 5. *Suppose $\mathcal{D} = (\Delta, H, V)$ is a domino system and $N > 1$. A* tiling *of $[N{\times}N]$ by $\mathcal{D}$ is a mapping $\tau : [N{\times}N] \to \Delta$ assigning to each square $(i, j) \in [N{\times}N]$ a tile type $\tau[i, j] \in \Delta$, subject to the following constraints:*

- *For each pair $(i, j), (i{+}1, j)$ of horizontally adjacent squares in $[N{\times}N]$, the corresponding pair $(\tau[i, j], \tau[i{+}1, j])$ of tile types satisfies H.*
- *For each pair $(i, j), (i, j{+}1)$ of vertically adjacent squares in $[N{\times}N]$, the corresponding pair $(\tau[i, j], \tau[i, j{+}1])$ of tile types satisfies V.*

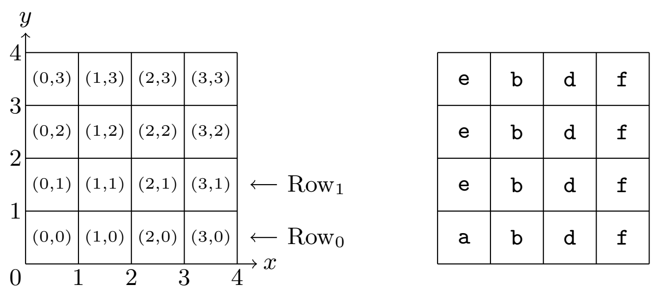

Fig. 2. The board $[4{\times}4]$ and one tiling of it by $\mathcal{D}_1$.

Example 4. Define a domino system $\mathcal{D}_1 = (\Delta, H, V)$ where

$$\Delta = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$$
$$H = \{(\mathtt{a}, \mathtt{b}), (\mathtt{b}, \mathtt{a}), (\mathtt{b}, \mathtt{d}), (\mathtt{c}, \mathtt{b}), (\mathtt{d}, \mathtt{c}), (\mathtt{d}, \mathtt{f}), (\mathtt{e}, \mathtt{b})\}$$
$$V = \{(\mathtt{a}, \mathtt{b}), (\mathtt{a}, \mathtt{e}), (\mathtt{b}, \mathtt{b}), (\mathtt{b}, \mathtt{c}), (\mathtt{c}, \mathtt{d}), (\mathtt{d}, \mathtt{d}), (\mathtt{e}, \mathtt{e}), (\mathtt{f}, \mathtt{f})\}.$$

The map $\tau : [4{\times}4] \to \Delta$ pictured in Figure 2 is a tiling of $[4{\times}4]$ by $\mathcal{D}_1$.

We need to be able to discuss partial tilings and tilings with initial conditions.

Definition 6. *Suppose $\mathcal{D} = (\Delta, H, V)$ is a domino system and $N > 1$.*

1. *Let $\mathbf{w} = (w_0, \ldots, w_{m-1}) \in \Delta^m$ with $0 < m \le N$, and let $j < N$. A tiling τ of $[N{\times}N]$ by $\mathcal{D}$* satisfies the initial condition $\mathbf{w}$ *if $\tau[i, 0] = w_i$ for all $i < m$.*
2. *If $U \subseteq [N{\times}N]$ then we may speak of tilings of U by $\mathcal{D}$ satisfying $\mathbf{w}$; these are mappings from U to Δ which satisfy those horizontal, vertical and initial condition constraints that mention squares in U only.*
3. *Given a tiling τ of $[N{\times}N]$ by $\mathcal{D}$, we say that τ* has a repeated row *if there exists $\mathbf{z} \in \Delta^N$ and distinct $j < k < N$ such that τ makes the same assignment to Row_j and to Row_k; that is, $\tau[i, j] = \tau[i, k]$ for all $0 \le i < N$.*

Example 3 (continued). The tiling of $[4{\times}4]$ pictured in Figure 2 satisfies the initial condition $(\mathtt{a}, \mathtt{b})$. However, $\mathcal{D}_1$ cannot tile $[4{\times}4]$ with initial condition $(\mathtt{b}, \mathtt{a})$.

In this paper we will be particularly interested in the following "exponential tiling problem," which we define in both local and uniform versions.

Definition 7. *1. Given a domino system $\mathcal{D} = (\Delta, H, V)$,* ExpTile$(\mathcal{D})$ *denotes the combinatorial decision problem whose input is a triple $(\mathcal{D}, m, \mathbf{w})$ where $m \ge 1$ and $\mathbf{w} \in \Delta^m$, and which asks whether $\mathcal{D}$ tiles $[2^m{\times}2^m]$ with initial condition $\mathbf{w}$.*
2. ExpTile $= \bigcup_{\mathcal{D}}$ ExpTile$(\mathcal{D})$.

3.1 A Domino System that Exponentially Counts

Our proofs of Theorems 1 and 2 will exploit the following fact.

Proposition 2. *There exists a domino system $\mathcal{D}_\mathrm{e} = (\Delta_\mathrm{e}, H_e, V_e)$ with the following property: for all $m > 2$ there exist m-tuples $\mathbf{w}_m, \mathbf{w}'_m \in (\Delta_\mathrm{e})^m$ such that*

1. *$\mathcal{D}_\mathrm{e}$ does not tile $[2^m{\times}2^m]$ with initial condition $\mathbf{w}_m$, but $\mathcal{D}_\mathrm{e}$ does tile U with initial condition $\mathbf{w}_m$ for every $U \subseteq [2^m{\times}2^m]$ satisfying $|U| < 2^m$.*
2. *$\mathcal{D}_\mathrm{e}$ tiles $[2^m{\times}2^m]$ with initial condition $\mathbf{w}'_m$, and moreover every tiling of $[2^m{\times}2^m]$ by $\mathcal{D}_\mathrm{e}$ with initial condition $\mathbf{w}'_m$ has no repeated row.*

We describe one way to construct such a domino system $\mathcal{D}_\mathrm{e}$. Our strategy is to design $\mathcal{D}_\mathrm{e}$ so that its tilings of subsets of $[2^m{\times}2^m]$ force consecutive rows to encode consecutive integers between 0 and $2^m{-}1$.

If $m > 0$ and $x \in \{0, 1, 2, 3, \ldots, 2^m - 1\}$, let $\mathrm{Bin}_m(x)$ denote the reverse m-bit binary representation of x (least significant bit at the left).

Example 5. $\mathrm{Bin}_5(6) = (0, 1, 1, 0, 0)$.

We define some sets of new symbols; they will be the tile types for $\mathcal{D}_e$:

$$\Delta_0 = \{0^-_L, 0^-_M, 0^+_M, 0^-_R, 0^+_R\} \qquad \Delta_1 = \{1^\diamond_L, 1^\diamond_M, 1^+_M, 1^\diamond_R, 1^+_R\}$$
$$\Delta_{01} = \Delta_0 \cup \Delta_1 \qquad \Delta_e = \Delta_{01} \cup \{\triangleleft\}.$$

Definition 8. *Suppose $m > 2$ and $x \in \{0, 1, 2, 3, \ldots, 2^m - 1\}$, with $\mathrm{Bin}_m(x) = (b_0, b_1, \ldots, b_{m-1})$. The annotated m-bit binary representation of x is the m-tuple $\mathrm{AnnBin}_m(x) = (\mathsf{a}_0, \mathsf{a}_1, \ldots, \mathsf{a}_{m-1}) \in (\Delta_{01})^m$ given as follows: $\mathsf{a}_i = (b_i)^s_X$ where*

- *X is L if $i = 0$, R if $i = m - 1$, and M otherwise.*
- *If there exists $j < i$ such that $b_j = 1$, then s is $+$. Otherwise, s is $-$ if $b_i = 0$ while s is $\diamond$ if $b_i = 1$.*

Example 6. $\mathrm{AnnBin}_5(6) = (0^-_L, 1^\diamond_M, 1^+_M, 0^+_M, 0^+_R)$.

Note that the "bases" of the entries of $\mathrm{AnnBin}_m(x)$ give the reverse m-bit binary representation of x; the subscripts are exactly $(L, M, \ldots, M, R)$; and the superscripts are one of the following patterns: $(\diamond, +, \ldots, +)$, $(-, \ldots, -, \diamond, +, \ldots, +)$, $(-, \ldots, -, \diamond)$, or $(-, -, \ldots, -)$, where $\diamond$ occurs at the first bit of x equalling 1.

Fix $m > 2$ and define τ_m to be the mapping $[2^m \times 2^m] \to \Delta_e$ which for each $0 \le j < 2^m$ assigns $\mathrm{AnnBin}_m(j)$ to the first m entries in Row_j, and assigns $\triangleleft$ to all remaining squares (see Figure 3).

Row_{15}	$1^\diamond_L$	1^+_M	1^+_M	1^+_R	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$
$\vdots$	$\vdots$															$\vdots$
Row_5	$1^\diamond_L$	0^+_M	1^+_M	0^+_R	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$
Row_4	0^-_L	0^-_M	$1^\diamond_M$	0^+_R	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$
Row_3	$1^\diamond_L$	1^+_M	0^+_M	0^+_R	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$
Row_2	0^-_L	$1^\diamond_M$	0^+_M	0^+_R	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$
Row_1	$1^\diamond_L$	0^+_M	0^+_M	0^+_R	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$
Row_0	0^-_L	0^-_M	0^-_M	0^-_R	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$	$\triangleleft$

Fig. 3. τ_4 defined on $[16\times16]$.

Now let $\mathcal{D}_e = (\Delta_e, H_e, V_e)$ be the smallest domino system with respect to which τ_4 is a tiling of $[16\times16]$. That is, define

$$H_e = \{0^-_L\}\times\{0^-_M, 1^\diamond_M\} \;\cup\; \{1^\diamond_L\}\times\{0^+_M, 1^+_M\} \;\cup\; \{0^-_M\}\times\{0^-_M, 1^\diamond_M, 0^-_R, 1^\diamond_R\}$$
$$\cup\; \{0^+_M, 1^+_M, 1^\diamond_M\}\times\{0^+_M, 1^+_M, 0^+_R, 1^+_R\} \;\cup\; \{0^-_R, 0^+_R, 1^\diamond_R, 1^+_R, \triangleleft\}\times\{\triangleleft\}$$
$$V_e = \{(0^-_L, 1^\diamond_L), (1^\diamond_L, 0^-_L), (0^-_M, 0^+_M), (0^+_M, 0^+_M), (0^+_M, 1^\diamond_M), (1^\diamond_M, 1^+_M), (1^+_M, 1^+_M),$$
$$(1^+_M, 0^-_M), (0^-_R, 0^+_R), (0^+_R, 0^+_R), (0^+_R, 1^\diamond_R), (1^\diamond_R, 1^+_R), (1^+_R, 1^+_R), (\triangleleft, \triangleleft)\}.$$

The reader can check that $\mathcal{D}_e$, thus defined, satisfies Proposition 2 with $\mathbf{w}_m = \mathrm{AnnBin}_m(1)$ and $\mathbf{w}'_m = \mathrm{AnnBin}_m(0)$. Indeed, τ_m is the unique tiling by $\mathcal{D}_e$ of $[2^m{\times}2^m]$ with initial condition $\mathbf{w}'_m$, and clearly τ_m has no repeated rows. On the other hand, $\mathcal{D}_e$ cannot tile $[2^m{\times}2^m]$ with initial condition $\mathbf{w}_m$ (as it cannot count past $2^m - 1$), but if $U \subseteq [2^m{\times}2^m]$ with $|U| < 2^m$, then there must exist $k < 2^m$ such that U is disjoint from Row_k. In this case $\mathcal{D}_e$ can easily tile U with initial condition $\mathbf{w}_m$, simply by assigning $\mathrm{AnnBin}_m(j{+}1)$ to the first m entries of Row_j for each $j < k$, assigning $\mathrm{AnnBin}_m(j)$ to the first m entries of Row_j for all $k < j < 2^m$, and $\triangleleft$ to all remaining entries.

4 Interpreting Exponential Tiling into Expressibility

In this section we will describe the main (and most difficult) construction of this paper. It takes as input an instance $(\mathcal{D}, m, \mathbf{w})$ of ExpTile where $m > 2$ and m is a power of 2, and produces as output an instance (D, Γ, R) of Expr, so that

$$R \text{ is expressible from } \Gamma \Leftrightarrow \mathcal{D} \text{ cannot tile } [2^m{\times}2^m] \text{ with initial condition } \mathbf{w}.$$

Furthermore, the existence of "small" witnesses to the expressibility or inexpressibility of R will be connected to the existence of "small" witnesses to untilability or tilability (small subsets of $[2^m{\times}2^m]$ that cannot be tiled, or tilings of $[2^m{\times}2^m]$ with repeated rows). Thus Proposition 2 will give us Theorems 1 and 2. Because we also wish the construction $(\mathcal{D}, m, \mathbf{w}) \mapsto (D, \Gamma, R)$ to give a logspace reduction of this fragment of ExpTile into $\neg$Expr, the sizes of D, Γ, and the relations in $\Gamma \cup \{R\}$ must be bounded by a polynomial in $|\Delta| + m$, and the construction itself must be executable in logspace in $|\Delta| + m$.

4.1 Defining the Domain D and Encoding $[2^m{\times}2^m]$ in D^m

For the remainder of Section 4 we fix a domino system $\mathcal{D} = (\Delta, H, V)$, an integer $m = 2^t$ $(t > 1)$, and an m-tuple $\mathbf{w} = (w_0, w_1, \ldots, w_{m-1}) \in \Delta^m$.

Definition 9. *The domain D for our constraint language is the disjoint union of the sets Δ, $P := \{\mathtt{p}_{00}, \mathtt{p}_{01}, \mathtt{p}_{10}, \mathtt{p}_{11}\}$, $\{0, 1\}$, $\{\mathtt{a}, \mathtt{b}\}$, $\{\top, \bot\}$, and $\{\infty\}$.*

We next explain how we will interpret $[2^m{\times}2^m]$ in D^m. For $(x, y) \in [2^m{\times}2^m]$, write $\mathrm{Bin}_m(x) = (x_0, x_1, \ldots, x_{m-1})$ and $\mathrm{Bin}_m(y) = (y_0, y_1, \ldots, y_{m-1})$, the reverse m-bit binary representations of x and y respectively, and let $\mathbf{p}(x, y) \in D^m$ be given by $\mathbf{p}(x, y)[i] = \mathtt{p}_{x_i y_i}$ for $0 \le i < m$. In this way the elements of $[2^m{\times}2^m]$ are put in one-to-one correspondence with the elements of P^m.

Example 7. If $m = 8$, then $\mathbf{p}(53, 188) = (\mathtt{p}_{10}, \mathtt{p}_{00}, \mathtt{p}_{11}, \mathtt{p}_{01}, \mathtt{p}_{11}, \mathtt{p}_{11}, \mathtt{p}_{00}, \mathtt{p}_{01})$.

Next we define $t + 1$ auxiliary elements $\beta_0, \beta_1, \ldots, \beta_{t-1}, \gamma$ in D^m (recall that $t = \log_2 m$), first by example. If $m = 8$ (so $t = 3$), then

$$\begin{aligned}
\beta_0 &= (0, 1, 0, 1, 0, 1, 0, 1) \\
\beta_1 &= (0, 0, 1, 1, 0, 0, 1, 1) \\
\beta_2 &= (0, 0, 0, 0, 1, 1, 1, 1) \\
\gamma &= (\mathtt{b}, \mathtt{b}, \mathtt{a}, \mathtt{b}, \mathtt{a}, \mathtt{a}, \mathtt{a}, \mathtt{b}).
\end{aligned}$$

Note that the columns on the right-hand side of the above equations, restricted to the β_i's, are $\mathrm{Bin}_3(0), \mathrm{Bin}_3(1), \mathrm{Bin}_3(2), \ldots, \mathrm{Bin}_3(7)$ respectively. In general,

Definition 10.

1. $\beta_0, \ldots, \beta_{t-1} \in \{0,1\}^m$ *are defined so that* $(\beta_0[i], \beta_1[i], \ldots, \beta_{t-1}[i]) = \mathrm{Bin}_t(i)$ *for all* $0 \le i < m$.
2. *The element* $\gamma \in \{\mathtt{a}, \mathtt{b}\}^m$ *is defined by* $\gamma[i] = \mathtt{b}$ *if* $i = 2^k - 1$ *for some* $k \le t$, *and* $\gamma[i] = \mathtt{a}$ *otherwise.*
3. $\mathbf{s} = (\beta_0, \beta_1, \ldots, \beta_{t-1}, \gamma) \in (D^m)^{t+1}$.
4. $R_0 = \mathrm{proj}(\mathbf{s}) = \{(\mathrm{Bin}_t(i), \gamma[i]) : 0 \le i < m\}$.

Example 8. If $m = 8$, then $R_0 = \{(0,0,0,\mathtt{b}), (1,0,0,\mathtt{b}), (0,1,0,\mathtt{a}), (1,1,0,\mathtt{b}), (0,0,1,\mathtt{a}), (1,0,1,\mathtt{a}), (0,1,1,\mathtt{a}), (1,1,1,\mathtt{b})\}$.

The elements $\beta_0, \ldots, \beta_{t-1}, \gamma \in D^m$ and the relation R_0 will help us coordinatize P^m. The element γ helps to enforce some "rigidity" as explained in the next lemma.

Lemma 1. *Suppose* σ *is a self-map from* $\{0, 1, \ldots, t-1\}$ *to itself, and* $\mathbf{d} = (\beta_{\sigma(0)}, \beta_{\sigma(1)}, \ldots, \beta_{\sigma(t-1)}, \gamma)$. *If* $\mathrm{proj}(\mathbf{d}) \subseteq R_0$, *then* $\sigma(i) = i$ *for all* $i < t$.

Once the constraint language Γ has been constructed, we will be intensely interested in the $(t+1)$-ary relation S expressed by $(\mathcal{I}_m(\Gamma), \mathbf{s})$. This relation is equivalently defined as the set of images of $(\beta_0, \ldots, \beta_{t-1}, \gamma)$ under the m-ary polymorphisms of Γ. We will be particularly interested in learning whether the $(t+1)$-tuple $(\top, \top, \ldots, \top)$ belongs to S. Call a map $f : D^m \to D$ *special* if it satisfies $f(\beta_0) = f(\beta_1) = \cdots = f(\beta_{t-1}) = f(\gamma) = \top$. The intermediate aim of the construction of Γ is to achieve the following two competing goals:

1. If $h : D^m \to D$ is any special m-ary polymorphism of Γ, then h should map P^m to Δ; moreover, the restriction of h to P^m should encode a tiling of $[2^m \times 2^m]$ by $\mathcal{D}$ with initial condition $\mathbf{w}$.
2. Conversely, if τ is any tiling by $\mathcal{D}$ of $[2^m \times 2^m]$ with initial condition $\mathbf{w}$, then there should exist a special m-ary polymorphism h of Γ whose restriction to P^m encodes τ.

An immediate consequence of these goals, when achieved, is that the expressible relation S will contain the constant tuple $(\top, \top, \ldots, \top)$ if and only if $\mathcal{D}$ tiles $[2^m \times 2^m]$ with initial condition $\mathbf{w}$. This will somehow help us in achieving the goals described at the beginning of Section 4.

4.2 Defining the Constraint Language Γ and the Test Relation R

Each relation in Γ will be constructed using the following recipe. Fix $k = 1$ or 2. Choose a k-ary relation $\mathcal{H}$ on P^m and a k-ary relation C on Δ, subject to the requirement that $\mathcal{H}$ factors as an m-fold product relation $\mathcal{H} = H_0 \times H_1 \times \cdots \times H_{m-1}$ for some k-ary relations $H_0, H_1, \ldots, H_{m-1}$ on P. Then define the $(k+t+1)$-ary relation $\mathcal{R}_{\mathcal{H} \Rightarrow C}$ on D as follows:

$$\mathcal{R}_{\mathcal{H}\Rightarrow C} = \bigcup_{i=0}^{m-1} \{(\mathbf{x},\mathbf{y}) \in P^k \times (\{0,1\}^t \times \{\mathtt{a},\mathtt{b}\}) \,:\, \mathbf{x} \in H_i,\ \mathbf{y} = (\mathrm{Bin}_t(i), \gamma[i])\}$$
$$\cup \{(\mathbf{x},\mathbf{y}) \in \Delta^k \times \{\top,\bot\}^{t+1} \,:\, \bot \in \{\mathbf{y}[0],\dots,\mathbf{y}[t]\} \text{ or } \mathbf{x} \in C\}$$
$$\cup \{(\infty,\infty,\dots,\infty)\}.$$

Lemma 2. *For any relation $\mathcal{R}_{\mathcal{H}\Rightarrow C}$ constructed according to the recipe above:*

1. $\mathcal{R}_{\mathcal{H}\Rightarrow C} \subseteq \big(P^k \times \{0,1\}^t \times \{\mathtt{a},\mathtt{b}\}\big) \ \cup\ \big(\Delta^k \times \{\top,\bot\}^{t+1}\big) \ \cup\ \{\infty\}^{k+t+1}$.
2. *For any* $\mathbf{c} \in (D^m)^k$, $\mathrm{proj}(\mathbf{c},\beta_0,\beta_1,\dots,\beta_{t-1},\gamma) \subseteq \mathcal{R}_{\mathcal{H}\Rightarrow C}$ *if and only if* $\mathbf{c} \in \mathcal{H}$.
3. *For any* $\mathbf{c} \in D^k$, $(\mathbf{c},\top,\top,\dots,\top) \in \mathcal{R}_{\mathcal{H}\Rightarrow C}$ *if and only if* $\mathbf{c} \in C$.

Our first family of relations will encode the adjacency constraints of $\mathcal{D}$.

Definition 11. *1. For an integer $0 < x < 2^m$ define $\lg(x)$ to be the largest integer $0 \le k < m$ such that 2^k divides x.*
2. For $0 \le k < m$ let $\mathcal{HA}^{(k)}, \mathcal{VA}^{(k)}$ be the following binary relations on P^m:

$$\mathcal{HA}^{(k)} = \{(\mathbf{p}(x,y),\mathbf{p}(x{+}1,y)) \,:\, 0 \le x,y < 2^m,\ x \ne 2^m{-}1,\ \lg(x{+}1) = k\}$$
$$\mathcal{VA}^{(k)} = \{(\mathbf{p}(x,y),\mathbf{p}(x,y{+}1)) \,:\, 0 \le x,y < 2^m,\ y \ne 2^m{-}1,\ \lg(y{+}1) = k\}.$$

I.e., $\mathcal{HA}^{(k)}$ is the binary relation on P^m encoding those pairs $((x,y),(x{+}1,y))$ of horizontally adjacent elements of $[2^m{\times}2^m]$ for which the reverse binary representation of x begins with k 1's followed by 0. The reader should verify that each of the relations $\mathcal{HA}^{(k)}, \mathcal{VA}^{(k)}$ factors as an m-fold product relation.

Example 9. If $m = 8$ and $k = 3$, then

$$\mathcal{HA}^{(3)} = \{(\mathtt{p}_{10},\mathtt{p}_{00}),(\mathtt{p}_{11},\mathtt{p}_{01})\}^3 \times \{(\mathtt{p}_{00},\mathtt{p}_{10}),(\mathtt{p}_{01},\mathtt{p}_{11})\}$$
$$\times \{(\mathtt{p}_{00},\mathtt{p}_{00}),(\mathtt{p}_{01},\mathtt{p}_{01}),(\mathtt{p}_{10},\mathtt{p}_{10}),(\mathtt{p}_{11},\mathtt{p}_{11})\}^4.$$

Definition 12. *Recall that $\mathcal{D} = (\Delta, H, V)$. The set of* adjacency relations *is*

$$\mathcal{A} \ = \ \{\mathcal{R}_{\mathcal{HA}^{(k)}\Rightarrow H} \,:\, 0 \le k < m\} \ \cup\ \{\mathcal{R}_{\mathcal{VA}^{(k)}\Rightarrow V} \,:\, 0 \le k < m\}.$$

For each $(x,y) \in [2^m{\times}2^m]$, the singleton unary relation $\{\mathbf{p}(x,y)\}$ on P^m clearly factors as an m-fold product relation.

Definition 13. *Recall that $\mathbf{w} = (w_0,\dots,w_{m-1})$. The set of* initial relations *is*

$$\mathcal{I} \ = \ \{\mathcal{R}_{\{\mathbf{p}(k,0)\}\Rightarrow\{w_k\}} \,:\, 0 \le k < m\}.$$

Definition 14. *Our constraint language is $\Gamma \ = \ \mathcal{A} \ \cup\ \mathcal{I} \ \cup\ \{\mathcal{R}_{P^n\Rightarrow\Delta}\}$.*

Finally, we define two further $(t+1)$-ary relations on D. The first relation, R, is an easily constructed relation whose expressibility from Γ will be our chief interest; it may be informally defined as $\mathcal{R}_{\top\Rightarrow\bot}$ where $\top$ and $\bot$ are here being used to denote the 0-ary "true" and "false" relations on P^m and Δ respectively. The second relation, S, is easily defined but not easily constructed and is not claimed to be part of the output of our logspace construction.

Definition 15. *Recall that* $R_0 = \mathrm{proj}(\mathbf{s})$ *where* $\mathbf{s} = (\beta_0, \beta_1, \ldots, \beta_{t-1}, \gamma)$.

$$R = R_0 \;\cup\; (\{\top, \bot\}^{t+1} \setminus \{(\top, \top, \ldots, \top)\}) \;\cup\; \{(\infty, \infty, \ldots, \infty)\}$$
$$S = \{(h(\beta_0), h(\beta_1), \ldots, h(\beta_{t-1}), h(\gamma)) \,:\, h \textit{ is an } m\textit{-ary polymorphism of } \Gamma\}.$$

4.3 Connecting Polymorphisms, Tilings, and Expressibility

For convenience, define the notation $\widehat{\top} = (\top, \top, \ldots, \top)$ and $\widehat{\infty} = (\infty, \infty, \ldots, \infty)$.

Lemma 3. *1. S is the smallest $(t+1)$-ary relation expressible from Γ and containing R_0.*
2. $R \subseteq S \subseteq R \cup \{\widehat{\top}\}$.
3. R is expressible from Γ if and only if $\widehat{\top} \notin S$.

Proof. $S = \pi_{\mathbf{s}}(\mathrm{Sol}(\mathcal{I}_m(\Gamma)))$, i.e., S is the relation expressed by $(\mathcal{I}_m(\Gamma), \mathbf{s})$ where $\mathbf{s} = (\beta_0, \ldots, \beta_{t-1}, \gamma)$. (1) follows from this observation, the definition of R_0, and Proposition 1. To prove $S \subseteq R \cup \{\widehat{\top}\}$, it thus suffices to show that $R \cup \{\widehat{\top}\}$ is expressible from Γ (as it clearly contains R_0). This is easy, since the primitive positive formula $\exists z \mathcal{R}_{P^m \Rightarrow \Delta}(z, x_0, x_1, \ldots, x_t)$ defines $R \cup \{\widehat{\top}\}$. As (3) follows from (1) and (2), it remains only to prove $R \subseteq S$.

Clearly $R_0 \subseteq S$ by (1), and $\widehat{\infty} \in S$ since the constant function $D^m \to \{\infty\}$ is a polymorphism of Γ. Suppose now that $\mathbf{f} = (f_0, \ldots, f_t) \in \{\top, \bot\}^{t+1} \setminus \{\widehat{\top}\}$. Pick any $d_0 \in \Delta$ and define $h_{\mathbf{f}} : D^m \to D$ by

$$h_{\mathbf{f}}(\mathbf{x}) \;=\; \begin{cases} d_0 \text{ if } \mathbf{x} \in P^m \\ f_i \text{ if } \mathbf{x} = \beta_i \text{ for some } i < t \\ f_t \text{ if } \mathbf{x} = \gamma \\ \bot \text{ if } \mathbf{x} \in \{0,1\}^m \cup \{\mathtt{a}, \mathtt{b}\}^m \setminus \{\beta_0, \ldots, \beta_{t-1}, \gamma\} \\ \infty \text{ otherwise.} \end{cases}$$

To prove $\mathbf{f} \in S$, it suffices to show that $h_{\mathbf{f}}$ is a polymorphism of Γ. We will show simply that $h_{\mathbf{f}}$ preserves each initial relation $\mathcal{R}_{\{\mathbf{p}(k,0)\} \Rightarrow \{w_k\}}$ at all $(t+2)$-tuples in D^m, the proofs for the other relations being similar. Indeed, if this were false, then there would exist $\mathbf{c} = (\mathbf{x}, \mathbf{z}_0, \ldots, \mathbf{z}_t) \in (D^m)^{t+2}$ with

(a) $\mathrm{proj}(\mathbf{c}) \subseteq \mathcal{R}_{\{\mathbf{p}(k,0)\} \Rightarrow \{w_k\}}$, but
(b) $(h_{\mathbf{f}}(\mathbf{x}), h_{\mathbf{f}}(\mathbf{z}_0), \ldots, h_{\mathbf{f}}(\mathbf{z}_t)) \notin \mathcal{R}_{\{\mathbf{p}(k,0)\} \Rightarrow \{w_k\}}$.

At least one of $h_{\mathbf{f}}(\mathbf{x}), h_{\mathbf{f}}(\mathbf{z}_0), \ldots, h_{\mathbf{f}}(z_t)$ must be different from ∞. Hence by definition of $h_{\mathbf{f}}$, $\{\mathbf{x}, \mathbf{z}_0, \ldots, \mathbf{z}_t\}$ is not disjoint from $P^m \cup \{0,1\}^m \cup \{\mathtt{a}, \mathtt{b}\}^m$. This last fact, Lemma 2(1), and item (a) above then yield $\mathbf{x} \in P^m$, $\mathbf{z}_0, \ldots, \mathbf{z}_{t-1} \in \{0,1\}^m$, and $\mathbf{z}_t \in \{\mathtt{a}, \mathtt{b}\}^m$. Hence $(h_{\mathbf{f}}(\mathbf{x}), h_{\mathbf{f}}(\mathbf{z}_0), \ldots, h_{\mathbf{f}}(\mathbf{z}_t)) = (d_0, f'_0, \ldots, f'_t)$ for some $f'_0, \ldots, f'_t \in \{\top, \bot\}$ (by the definition of $h_{\mathbf{f}}$). If $d_0 = w_k$ or at least one f'_i is $\bot$, then clearly $(d_0, f'_0, \ldots, f'_t) \in \mathcal{R}_{\{\mathbf{p}(k,0)\} \Rightarrow \{w_k\}}$; hence $d_0 \neq w_k$ and all f'_i are $\top$. The definition of $h_{\mathbf{f}}$ then implies that $\mathbf{z}_t = \gamma$ and there exists a selfmap σ on $\{0, 1, \ldots, t-1\}$ such that $\mathbf{z}_i = \beta_{\sigma(i)}$ for $i < t$. Lemma 1 then implies that $\sigma(i) = i$ for all $i < t$, so $\mathbf{c} = (\mathbf{x}, \beta_0, \ldots, \beta_{t-1}, \gamma)$ with $\mathbf{x} \in P^m$. The definition of $h_{\mathbf{f}}$ then gives $(d_0, \top, \ldots, \top) = (d_0, f_0, \ldots, f_t)$, contradicting the assumption that $\mathbf{f} \neq \widehat{\top}$. □

We can now prove the desired connection between tilings and expressibility.

Proposition 3. *The following are equivalent:*

1. *R is not expressible from Γ.*
2. *$\widehat{\top} \in S$.*
3. *$\mathcal{D}$ tiles $[2^m \times 2^m]$ with initial condition $\mathbf{w}$.*

Proof. (1) ⇔ (2) follows from Lemma 3.

(2) ⇒ (3). Assume $\widehat{\top} \in S$; choose an m-ary polymorphism h of Γ satisfying $h(\beta_0) = \cdots = h(\beta_{t-1}) = h(\gamma) = \top$. We first show that h maps P^m into Δ. Indeed, let $\mathbf{x} \in P^m$; then $\text{proj}((\mathbf{x}, \beta_0, \ldots, \beta_{t-1}\gamma)) \subseteq \mathcal{R}_{P^m \Rightarrow \Delta}$ by Lemma 2(2). As h is a polymorphism of Γ, it preserves $\mathcal{R}_{P^m \Rightarrow \Delta}$ at $(\mathbf{x}, \beta_0, \ldots, \beta_{t-1}, \gamma)$; hence we get $(h(\mathbf{x}), h(\beta_0), \ldots, h(\beta_{t-1}), h(\gamma)) \in \mathcal{R}_{P^m \Rightarrow \Delta}$, i.e., $(h(\mathbf{x}), \top, \ldots, \top) \in \mathcal{R}_{P^m \Rightarrow \Delta}$. This with Lemma 2(3) implies $h(\mathbf{x}) \in \Delta$, as claimed.

Thus we may define a map $\tau_h : [2^m \times 2^m] \to \Delta$ by $\tau_h[i,j] = h(\mathbf{p}(i,j))$. Using the fact that h preserves the adjacency and initial relations at all tuples of the form $(\mathbf{x}, \mathbf{x}', \beta_0, \ldots, \beta_{t-1}, \gamma)$ or $(\mathbf{x}, \beta_0, \ldots, \beta_{t-1}, \gamma)$ respectively ($\mathbf{x}, \mathbf{x}'$ varying over P^m), and using Lemma 2(2,3), one can show that τ_h is a tiling of $[2^m \times 2^m]$ with initial condition $\mathbf{w}$.

(3) ⇒ (2). Assume that τ is a tiling of $[2^m \times 2^m]$ by $\mathcal{D}$ with initial condition $\mathbf{w}$. Define $h_\tau : D^m \to D$ by

$$h_\tau(\mathbf{x}) \;=\; \begin{cases} \tau[i,j] & \text{if } \mathbf{x} = \mathbf{p}(i,j) \text{ where } (i,j) \in [2^m \times 2^m] \\ \top & \text{if } \mathbf{x} \in \{\beta_0, \ldots, \beta_{t-1}, \gamma\} \\ \bot & \text{if } \mathbf{x} \in \{0,1\}^m \cup \{\mathtt{a}, \mathtt{b}\}^m \setminus \{\beta_0, \ldots, \beta_{t-1}, \gamma\} \\ \infty & \text{otherwise.} \end{cases}$$

It suffices to prove that h_τ is a polymorphism of Γ. We repeat the proof that $h_{\mathbf{f}}$ preserves $\mathcal{R}_{\{\mathbf{p}(k,0)\} \Rightarrow \{w_k\}}$ in the proof of Lemma 3, replacing $h_{\mathbf{f}}$ with h_τ. Again, we suppose for the sake of contradiction that we have $\mathbf{c} = (\mathbf{x}, \mathbf{z}_0, \ldots, \mathbf{z}_t) \in (D^m)^{t+2}$ with

(a) $\text{proj}(\mathbf{c}) \subseteq \mathcal{R}_{\{\mathbf{p}(k,0)\} \Rightarrow \{w_k\}}$, but
(c) $(h_\tau(\mathbf{x}), h_\tau(\mathbf{z}_0), \ldots, h_\tau(\mathbf{z}_t)) \notin \mathcal{R}_{\{\mathbf{p}(k,0)\} \Rightarrow \{w_k\}}$.

Arguing as before, we get

(d) $\mathbf{c} = (\mathbf{x}, \beta_0, \ldots, \beta_{t-1}, \gamma)$, and
(e) $\mathbf{x} \in P^m$ and $h_\tau(\mathbf{x}) \neq w_k$.

Items (a) and (d), with Lemma 2, imply $\mathbf{x} = \mathbf{p}(k,0)$. Hence $h_\tau(\mathbf{x}) = \tau[k,0]$, which with item (e) contradicts the fact that τ satisfies $\mathbf{w}$ at Row_0. □

As $|R| = 3m$, Corollary 1 implies that if R is expressible from Γ then R can be expressed by a CSP(Γ) instance having $|D|^{3m}$ variables, while if R is not expressible from Γ then this is witnessed by a polymorphism of Γ of arity $3m$. We can slightly improve this. On the one hand, Lemma 3 clearly implies:

Corollary 2. *If R is not expressible from Γ, then this is witnessed by an m-ary polymorphism.*

Conversely, a careful examination of the proof of Proposition 3(2)⇒(3) shows that the only constraints on h needed to complete the proof are ones involving the values of h at elements of $P^m \cup \{\beta_0, \ldots, \beta_{t-1}, \gamma\}$. Hence:

Corollary 3. *If R is expressible from Γ, then it can be expressed by an instance of* CSP(Γ) *(or a primitive positive formula over Γ) with $2^{2m} + t + 1$ variables.*

4.4 Refining Proposition 3

Proposition 4. *Suppose R is not expressible from Γ and this is witnessed by some polymorphism of Γ of arity $k < m$. Then there exists a tiling τ of $[2^m \times 2^m]$ by $\mathcal{D}$ with initial condition $\mathbf{w}$ with the property that every row of τ is repeated.*

Proof. Let h be the k-ary polymorphism of Γ; choose $\mathbf{c} = (\alpha_0, \alpha_1, \ldots, \alpha_t) \in (D^k)^{t+1}$ such that $\mathrm{proj}(\mathbf{c}) \subseteq R$ but $(h(\alpha_0), \ldots, h(\alpha_t)) \notin R$. Since S *is* expressible from Γ, h preserves S at $\mathbf{c}$, so $(h(\alpha_0), \ldots, h(\alpha_t)) \in S$. As $S \setminus R = \{\widehat{\top}\}$, we get $h(\alpha_i) = \top$ for all $i \leq t$.

For each $1 \leq i \leq k$ let $\mathbf{c}_i = (\alpha_0[i], \ldots, \alpha_t[i]) \in R$. Define

$$M = \{i : \mathbf{c}_i \in R_0\}$$
$$Q = \{i : \mathbf{c}_i \in \{\top, \bot\}^{t+1} \setminus \{\widehat{\top}\}\}$$
$$Z = \{i : \mathbf{c}_i = \widehat{\infty}\}.$$

For each $i \in M$, define $\sigma(i)$ to be the unique $j \in \{0, 1, \ldots, m-1\}$ such that $\mathbf{c}_i = (\beta_0[j], \ldots, \beta_{t-1}[j], \gamma[j])$. Now define a map $\lambda : [2^m \times 2^m] \to D^k$ as follows: given $(x, y) \in [2^m \times 2^m]$ and $1 \leq i \leq k$,

$$\lambda(x,y)[i] \;=\; \begin{cases} \mathbf{p}(x,y)[j] & \text{if } i \in M \text{ and } \sigma(i) = j \\ \top & \text{if } i \in Q \\ \infty & \text{if } i \in Z. \end{cases}$$

We will use λ to "represent" the elements of $[2^m \times 2^m]$ as elements of D^k (though we will see below that λ is not injective). We now loosely follow the proof of Proposition 3(2)⇒(3). Suppose $(x, y) \in [2^m \times 2^m]$ and let $\mathbf{x} = \lambda(x, y)$. One can check that $\mathrm{proj}(\mathbf{x}, \alpha_0, \ldots, \alpha_t) \subseteq \mathcal{R}_{P^m \Rightarrow \Delta}$. As h is a polymorphism, this implies $(h(\mathbf{x}), h(\alpha_0), \ldots, h(\alpha_t)) \in \mathcal{R}_{P^m \Rightarrow \Delta}$, i.e., $(h(\mathbf{x}), \top, \ldots, \top) \in \mathcal{R}_{P^m \Rightarrow \Delta}$. Hence $h(\mathbf{x}) \in \Delta$. Thus we may define a map $\tau_h : [2^m \times 2^m] \to \Delta$ by $\tau_h[x, y] = h(\lambda(x, y))$. As in the proof of Proposition 3(2⇒3), it will follow that τ_h is a tiling of $[2^m \times 2^m]$ by $\mathcal{D}$ with initial condition $\mathbf{w}$.

Observe that $|M| \leq k < m$, so the map σ is not surjective. Pick some $0 \leq j < m$ with $j \notin \mathrm{range}(\sigma)$. Then the map λ has the property that if $x, x', y, y' \in \{0, 1, \ldots, 2^m - 1\}$ and the binary representations of x and x' (y and y') agree everywhere except at bit j, then $\lambda(x, y) = \lambda(x', y')$. The same must therefore be true of the tiling τ_h. Hence every row (and every column) of τ_h is repeated. □

Proposition 5. *Suppose R can be expressed from Γ by an instance of* CSP(Γ) *(or primitive positive formula) with $k < 2^{2m}$ variables. Then there exists a subset $U \subseteq [2^m \times 2^m]$ with $|U| \leq k$ such that $\mathcal{D}$ does not tile U with initial condition $\mathbf{w}$.*

Proof. Choose an instance $\mathcal{P} = (X, D, \mathcal{C})$ of CSP(Γ) and a $(t+1)$-tuple $\mathbf{s} = (s_0, \ldots, s_t)$ of variables from X such that $(\mathcal{P}, \mathbf{s})$ expresses R and $|X| = k$. Thus

$$R = \{(h(s_0), \ldots, h(s_t)) \,:\, h \in \mathrm{Sol}(\mathcal{P})\}. \tag{1}$$

For each $h \in \mathrm{Sol}(\mathcal{P})$ define $\mathbf{c}_h = (h(s_0), \ldots, h(s_t)) \in R$. Define

$$\begin{aligned} M &= \{h \in \mathrm{Sol}(\mathcal{P}) \,:\, \mathbf{c}_h \in R_0\} \\ Q &= \{h \in \mathrm{Sol}(\mathcal{P}) \,:\, \mathbf{c}_h \in \{\top, \bot\}^{t+1} \setminus \{\widehat{\top}\}\} \\ Z &= \{h \in \mathrm{Sol}(\mathcal{P}) \,:\, \mathbf{c}_h = \widehat{\infty}\}. \end{aligned}$$

Next define

$$\mathcal{A} = \{x \in X \,:\, [h(x) \in P \ \forall h \in M] \ \&\ [h(x) \in \Delta \ \forall h \in Q] \ \&\ [h(x) = \infty \ \forall h \in Z]\}.$$

Similarly, define $\mathcal{B}$ to be the set of all $x \in X$ whose values under h in M, Q, Z are in $\{0, 1\}$, $\{\top, \bot\}$ and $\{\infty\}$ respectively; and define $\mathcal{E}$ to be the set of all $x \in X$ whose values under h in M, Q, Z are in $\{\mathtt{a}, \mathtt{b}\}$, $\{\top, \bot\}$ and $\{\infty\}$ respectively;

For each $0 \leq i < m$ choose $h_i \in M$ so that $(h_i(s_0), \ldots, h_i(s_t)) = (\mathrm{Bin}_t(i), \gamma[i])$. (Such h_i must exist by equation 1.) Now define $\lambda : \mathcal{A} \to P^m$ as follows: for $x \in \mathcal{A}$ and $0 \leq i < m$, put $\lambda(x)[i] = h_i(x)$.

Define $U = \{(i,j) \in [2^m \times 2^m] \,:\, \mathbf{p}(i,j) \in \mathrm{range}(\lambda)\}$. Clearly $|U| \leq |\mathcal{A}| \leq |X| = k$. We claim that $\mathcal{D}$ cannot tile U with initial condition $\mathbf{w}$. Assume to the contrary that $\tau : U \to \Delta$ is such a tiling. Define $h_\tau : X \to \Delta$ by

$$h_\tau(x) \ = \ \begin{cases} \tau[i,j] & \text{if } x \in \mathcal{A} \text{ and } \lambda(x) = \mathbf{p}(i,j) \\ \top & \text{if } x = s_j \text{ for some } 0 \leq j \leq t \\ \bot & \text{if } x \in \mathcal{B} \cup \mathcal{E} \setminus \{s_0, \ldots, s_t\} \\ \infty & \text{otherwise.} \end{cases}$$

It can be shown, essentially following the proof of Proposition 3(3⇒2), that h_τ is a solution of $\mathcal{P}$. But this with the fact that $(h_\tau(s_0), \ldots, h_\tau(s_t)) = \widehat{\top} \notin R$ contradicts equation 1. □

5 Conclusion

Proof of Theorem 1. Given $n = 3m$ where $m = 2^t$, $t > 1$, take $\mathcal{D}_\mathrm{e}$ and $\mathbf{w}_m$ as in Proposition 2(1), and let (D, Γ_n, R_n) be the output of our construction on input $(\mathcal{D}_\mathrm{e}, m, \mathbf{w}_m)$. (Note that D is independent of n, and $|D| = 22$ if we use the specific domino system $\mathcal{D}_\mathrm{e}$ described in Subsection 3.1.) We have $|R_n| = 3m = n$. By Proposition 3, R_n is expressible from Γ_n but, by Proposition 5, not by any CSP(Γ_n) instance having fewer than 2^m variables. □

Proof of Theorem 2. Follows similarly from Propositions 2(2), 3 and 4. □

Proof sketch of Theorem 3. Let $\text{ExpTile}_2(\mathcal{D})$ be the restriction of $\text{ExpTile}(\mathcal{D})$ to inputs $(\mathcal{D}, m, \mathbf{w})$ where $m = 2^t$, $t > 1$. Standard modifications of the proof of [2, Theorem 6.1.2], replacing the torus with the plane as in [10], show that every problem $\mathcal{P} \in$ **NEXPTIME** has a logspace reduction to $\text{ExpTile}_2(\mathcal{D})$ for some domino system $\mathcal{D}$. Via a "universal domino system" argument we can get a single domino system $\mathcal{D}_{\mathtt{u}} = (\Delta_{\mathtt{u}}, H_{\mathtt{u}}, V_{\mathtt{u}})$ such that $\text{ExpTile}_2(\mathcal{D}_{\mathtt{u}})$ is **NEXPTIME**-complete. Let $d = |\Delta_{\mathtt{u}}| + 11$. Our construction and Proposition 3 give a logspace reduction of $\text{ExpTile}_2(\mathcal{D}_{\mathtt{u}})$ to the restriction of $\neg\text{Expr}$ to d-element domains. □

We end with two questions.

1. Can d in Theorem 3 be reduced to $d = 3$, confirming the AIM conjecture?
2. Can Theorems 1–3 be improved so that both the domain *and* the constraint language are fixed and only the test relation varies? (Such an improvement of Theorem 3 would complement a result of Kozik for functions [16].)

References

1. Bodnarčuk, V. G., Kalužnin, L. A., Kotov, V. N., Romov, B. A.: Galois theory for Post algebras. I. Cybernetics and Systems Analysis 5, 243–252 (1969)
2. E. Börger, E. Grädel and Y. Gurevich, The Classical Decision Problem, Springer, Heidelberg (1997)
3. Bulatov, A., Krokhin, A., Jeavons, P.: Classifying the complexity of constraints using finite algebras. SIAM J. Comput. 34, 720–742 (2005)
4. ten Cate, B.: Notes on AIM CSP workshop, April 21, 2008, `http://www.aimath.org/WWN/constraintsatis/constraintsatis.pdf`
5. Cohen, D., Jeavons, P.: Tractable constraint languages. In Dechter, R., Constraint Processing, pp. 299–331. Elsevier, San Francisco (2003)
6. Creignou, N., Kolaitis, P., Zanuttini, B.: Structure identification of boolean relations and plain bases for co-clones. J. Comput. System Sci. 74, 1103-1115 (2008)
7. Dalmau, V.: Computational complexity of problems over generalized formulas. PhD thesis, Universitat Politécnica de Catalunya (2000)
8. Dechter, R., Pearl, J.: Structure identification in relational data. Artificial Intelligence 58, 237–270 (1992)
9. Geiger, D.: Closed systems of functions and predicates. Pacific J. Math. 27, 95–100 (1968)
10. Grädel, E.: Dominoes and the complexity of subclasses of logical theories. Ann. Pure Appl. Logic 43, 1–30 (1989)
11. Jeavons, P.: Constructing constraints. In: Maher, M., Puget, J.-F. (eds.) CP 1998. LNCS 1520, pp. 2–16. Springer, Heidelberg (1998)
12. Jeavons, P.: On the algebraic structure of combinatorial problems. Theoret. Comput. Sci. 200, 185–204 (1998)
13. Jeavons, P.: Presenting constraints. In: Giese, M., Waakerm A. (eds.) TABLEAUX 2009. LNAI, vol. 5607, pp. 1–15. Springer, Heidelberg (2009)
14. Jeavons, P., Cohen, D., Gyssens, M.: A test for tractability. In: Freuder, E. C. (ed.) CP 1996. LNCS 1118, pp. 267–281. Springer, Heidelberg (1996)
15. Jeavons, P., Cohen, D., Gyssens, M.: How to determine the expressive power of constraints. Constraints 4, 113–131 (1999)
16. Kozik, M.: A finite set of functions with an EXPTIME-complete composition problem. Theoret. Comput. Sci. 407, 330–341 (2008)