The Expressive Rate of Constraints

Hubie Chen*

Abstract

In reasoning tasks involving logical formulas, high expressiveness is desirable, although it often leads to high computational complexity. We study a simple measure of expressiveness: the number of formulas expressible by a language, up to semantic equivalence. We prove a dichotomy theorem on constraint languages regarding this measure.

1 Introduction

In reasoning tasks involving logical formulas, there is a trade-off between computational complexity and the expressiveness of the language in which the formulas are specified. Consequently, much effort has been directed towards identifying language restrictions which yield tractable reasoning. In many cases, these language restrictions, which are typically defined syntactically, impose semantic restrictions on the sets of models that can be expressed. For instance, it is known that a class of propositional models is expressible as a Horn formula if and only if it is closed under the coordinate-wise boolean *AND* operation.

A basic question thus arises: how *expressive* are various languages for specifying knowledge? That is, how rich are the classes of models that can be expressed by different languages? Among different languages yielding tractable reasoning, those that are most expressive will be of the greatest utility in many contexts. Consider the following two examples, from the area of *knowledge compilation* (see [2] for a survey). If one wants to preprocess a knowledge base specified in an intractable language into one specified in a tractable language, to perform rapid on-line query processing, the most expressive tractable languages are those that are most likely to be applicable. Or, if one wants to approximate a knowledge base specified in an intractable language by one (or more) knowledge bases specified in a tractable language, the most expressive tractable languages are those most likely to yield close approximations.

In this paper, we consider the most simple and fundamental measure of expressiveness that we can conceive of: we grade languages according to the *number* of different formulas that they can express – up to semantic equivalence. That is, we consider the number of different classes of models expressible by a language. We study expressiveness in the context of constraint languages, sets of relations which can be used to specify constraint networks. The notion of constraint language has been used heavily to identify tractable cases of the constraint satisfaction problem (CSP) and

*Department of Computer Science, Cornell University, Ithaca, NY 14853, USA. Email: hubes@cs.cornell.edu.

variants thereof [3], and can be used to describe restricted classes of propositional formulas such as Horn formulas and 2-SAT formulas.

The precise notion of expressibility we consider is asymptotic in nature: for a constraint language Γ, we consider the number of relations of arity n expressible by Γ (denoted by $||\Gamma||_n$), as n tends to infinity. On a finite domain A, the number of possible tuples of an n-ary relation is $|A|^n$, and so the number of different n-ary relations is equal to $2^{|A|^n}$, which upper bounds $||\Gamma||_n$ by definition. Our main result is a dichotomy theorem on constraint languages over a two-element domain, showing that all such constraint languages Γ are either *polynomially expressive* – that is, such that $\log ||\Gamma||_n$ is polynomial in n – or, *exponentially expressive* – that is, such that $\log ||\Gamma||_n$ is $\Omega(b^n)$, for some $b \geq 1$. It is worth emphasizing that the existence of such a dichotomy cannot be taken for granted in light of the many functions having growth rates that are intermediate between those that we identify – for instance, $n^{\log n}$ is neither polynomial nor exponential (according to our definitions).[1]

In order to study the expressivity of constraint languages, we make use of the algebraic viewpoint on constraint languages that has been used to study the complexity of the CSP [7, 4]. We demonstrate a number of sufficient conditions for showing polynomial or exponential expressivity that apply to constraint languages over finite domains of all sizes. To establish our dichotomy theorem, we use these sufficient conditions in conjunction with Post's classification theorem on function classes.

We remark that our dichotomy theorem has an alternative, rather mathematical formulation in terms of algebras. Namely, the theorem can be stated as a dichotomy theorem characterizing, for each algebra $\mathcal{A}$ with two elements, the asymptotics of the number of subalgebras $S_n(\mathcal{A})$ of the n-fold direct product of $\mathcal{A}$. Precisely, our dichotomy theorem implies that $\log S_n(\mathcal{A})$ is always polynomial or exponential in n (for two-element algebras $\mathcal{A}$).

2 Preliminaries

We use the notation $[n]$ as shorthand for the set $\{1, \ldots, n\}$. We use t_i to denote the ith coordinate of a tuple $\overline{t}$.

2.1 Constraints

Definition 2.1 *A relation of arity $k \geq 1$ over A is a subset of A^k. A constraint over A is an expression of the form $R(\overline{v})$ where R is a relation over A and $\overline{v}$ is a tuple of variables with the same arity as R.*

As usual, we consider an arity k constraint $R(\overline{v})$ to be satisfied by an assignment f defined on the variables in $\overline{v}$ if $(f(v_1), \ldots, f(v_k)) \in R$. Note that in this paper, we will only be concerned with constraints over finite domains A of nontrivial size, that is, of size greater than or equal to two.

[1] It has similarly been pointed out that the existence of complexity dichotomy theorems such as Schaefer's theorem [8] are not completely unsurprising in light of Ladner's theorem, which implies the existence of sets of intermediate complexity between P and NP, under the assumption P does not equal NP.

Definition 2.2 *A* constraint language *is a set of relations, all of which have the same domain. A constraint language is said to be* finite *if it contains a finite number of relations.*

Definition 2.3 *When* Γ *is a constraint language, a* Γ-formula *is defined to be an expression of the form*

$$\exists y_1, \ldots, y_m \, \mathcal{C}(x_1, \ldots, x_n, y_1, \ldots y_m)$$

where $\mathcal{C}$ *is a finite conjunction of constraints, each of which has relation from* $\Gamma \cup \{=_A\}$ *and variables from the set* $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$*. The variables* $\{x_1, \ldots, x_n\}$ *are said to be the* free variables *of the formula.*

By $=_A$, we denote the equality relation $\{(a, a) : a \in A\}$ on the set A.

The CSP over a particular constraint language can be defined in the following way.

Definition 2.4 *Let* Γ *be a finite constraint language over a finite domain* A*. The CSP*(Γ) *decision problem is to decide, given as input a* Γ*-formula* ϕ *with no free variables, whether or not* ϕ *is true.*

2.2 Expressivity

We formalize the class of relations expressible by a constraint language Γ in the following way.

Definition 2.5 *We say that an arity* k *relation* R *is* expressible *by the constraint language* Γ *if there exists a* Γ*-formula* ϕ *with free variables* $\{x_1, \ldots, x_k\}$ *such that* $R(x_1, \ldots, x_k)$ *is equivalent to* ϕ*, by which we mean that an assignment* $f : \{x_1, \ldots, x_k\} \to A$ *satisfies* $R(x_1, \ldots, x_k)$ *if and only if* ϕ *is true under* f*. We let* $\langle\Gamma\rangle$ *denote the set of all relations expressible by* Γ*.*

Example 2.6 *Define* Γ *to be the finite constraint language* $\{R, O, Z\}$ *over the domain* $\{0, 1\}$ *where* R *is the arity three relation* $\{0, 1\}^3 \setminus \{(1, 1, 0)\}$*,* O *is the arity one relation* $\{(0)\}$*, and* Z *is the arity one relation* $\{(1)\}$*. Every relation induced by a Horn clause can be expressed by* Γ*. For example, consider the relation* $S = \{(a_1, a_2, a_3, a_4) \in \{0, 1\}^4 : a_1 \wedge a_2 \wedge a_3 \Rightarrow a_4\}$*. We have that* $S(x_1, x_2, x_3, x_4)$ *is equivalent to the* Γ*-formula* $\exists y(R(x_1, x_2, y) \wedge R(y, x_3, x_4))$*. Notice that the relation* R *is equal to the set* $\{(a_1, a_2, a_3) \in \{0, 1\}^3 : a_1 \wedge a_2 \Rightarrow a_3\}$*. In fact, it can be shown that for any relation* T*, the formula* $T(x_1, \ldots, x_k)$ *is equivalent to a Horn formula without existentially quantified variables if and only if* T *is in* $\langle\Gamma\rangle$*.*

Definition 2.7 *When* Γ *is a constraint language, let* $||\Gamma||_n$ *denote the number of relations in* $\langle\Gamma\rangle$ *of arity* n*.*

Observe that when the relations in Γ are over a finite set A, $||\Gamma||_n$ is bounded above by $2^{|A|^n}$. Our interest is in differentiating between two modes of growth in the functions $||\Gamma||_n$, formalized in the following definitions.

Definition 2.8 *We say that a function* $f(n)$ *(assumed to have the natural numbers as both domain and range) is* polynomial *in* n *if* $f(n)$ *is* $O(n^k)$ *for some* $k \geq 1$*; we say that* $f(n)$ *is* exponential *in* n *if* $f(n)$ *is* $\Omega(b^n)$ *for some* $b > 1$*.*

Definition 2.9 *A constraint language* Γ *is* polynomially expressive *if* $\log ||\Gamma||_n$ *is polynomial in* n*, and is* exponentially expressive *if* $\log ||\Gamma||_n$ *is exponential in* n*.*

2.3 Algebra

In this section, we describe the dual algebraic viewpoint on constraint languages that we will use to study expressivity. This viewpoint has been used to study CSP complexity; we refer the reader to [7, 4] for more information.

Definition 2.10 *When A is a set and $k \geq 1$ is a natural number, a mapping $f : A^k \to A$ is called an* operation on A *of* rank k.

We let $\mathcal{O}_A$ denote the set of all operations on A of finite rank, and let $\mathcal{R}_A$ denote the set of all relations on A of finite arity.

Definition 2.11 *A rank k operation $f \in \mathcal{O}_A$* preserves *an arity n relation $R \in \mathcal{R}_A$ if for all tuples $\overline{t_1}, \ldots, \overline{t_k}$ in R, the tuple*

$$f(\overline{t_1}, \ldots, \overline{t_k}) \stackrel{\text{def}}{=} (f(t_{11}, \ldots, t_{k1}), \ldots, f(t_{1n}, \ldots, t_{kn}))$$

is also in R.

Definition 2.12 *For a set of operations $F \subseteq O_A$, define*

$$\mathit{Inv}(F) = \{R \in \mathcal{R}_A : R \textit{ is preserved by every operation in } F\}.$$

For a constraint language $\Gamma \subseteq R_A$, define

$$\mathit{Pol}(\Gamma) = \{f \in \mathcal{O}_A : f \textit{ preserves every relation in } \Gamma\}.$$

The Inv$(\cdot)$ and Pol$(\cdot)$ operators, in fact, form a Galois connection [6]. These operators are connected to the class of relations expressible by a constraint language in the following way.

Theorem 2.13 *[4] For any set of relations $\Gamma \subseteq \mathcal{R}_A$, $\langle\Gamma\rangle = \mathit{Inv}(\mathit{Pol}(\Gamma))$.*

From this theorem, Jeavons [4] demonstrated that the complexity of CSP(Γ) is an invariant of Pol(Γ) – that is, any two constraint languages Γ_1, Γ_2 such that Pol$(\Gamma_1) =$ Pol(Γ_2) give rise to problems CSP(Γ_1), CSP(Γ_2) having exactly the same complexity. Similarly, we can deduce from this theorem that the expressive rate of a constraint language Γ is an invariant of Pol(Γ).

Corollary 2.14 *For any sets of relations $\Gamma_1, \Gamma_2 \subseteq \mathcal{R}_A$, if $\mathit{Pol}(\Gamma_1) = \mathit{Pol}(\Gamma_2)$, then $\|\Gamma_2\|_n = \|\Gamma_1\|_n$ (for all $n \geq 1$).*

In other words, the expressivity of Γ depends *only* on Pol(Γ). Indeed, our program of classifying the constraint languages Γ according to their expressivity can be rephrased as a program of classifying the various sets of operations Pol(Γ) according to their expressivity, which we define in the following way.

Definition 2.15 *When $F \subseteq \mathcal{O}_A$ is a set of operations, we say that F is* polynomially expressive *(*exponentially expressive*) if $\mathit{Inv}(F)$ is polynomially expressive (exponentially expressive).*

The key feature of this definition is the following proposition.

Proposition 2.16 *A constraint language $\Gamma \subseteq \mathcal{R}_A$ is polynomially expressive (exponentially expressive) if and only if Pol(Γ) is polynomially expressive (exponentially expressive).*

The approach that we take in this paper is to prove results on the expressiveness of constraint languages Γ by studying the sets of operations having the form Pol(Γ). Fortunately, such sets of operations have a particular structure: they are *clones*.

Definition 2.17 *A set of operations $F \subseteq \mathcal{O}_A$ is a* clone *on A if the following two conditions hold.*

- *F contains all* projections*, that is, operations of the form $e^i_n(x_1, \ldots, x_n) = x_i$ (with $1 \leq i \leq n$).*
- *F is closed under* composition*, where the composition of a rank n operation $f \in \mathcal{O}_A$ and rank m operations $g_1, \ldots, g_n \in O_A$ is defined to be the rank m operation*

$$f(g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots, x_m)).$$

Proposition 2.18 *For every constraint language $\Gamma \subseteq \mathcal{R}_A$, the set of operations Pol(Γ) is a clone.*

We remark that studying the expressiveness of constraint languages is equivalent to studying an asymptotic question concerning algebras. The number of subalgebras of the n-fold direct product of an algebra (A, F) is exactly $||\text{Inv}(F)||_n$. Moreover, the expressive rate $||\Gamma||_n$ of a constraint language is the number of subalgebras of the n-fold direct product of the algebra $(A, \text{Pol}(F))$.

Proposition 2.19 *Let (A, F) be an algebra, that is, a domain A paired with a set of operations $F \subseteq \mathcal{O}_A$. A relation $R \subseteq A^n$ is the universe of a subalgebra of $(A, F)^n$ if and only if $R \in$ Inv(F).*

Proposition 2.20 *Let $\Gamma \subseteq \mathcal{O}_A$ be a constraint language. A relation $R \subseteq A^n$ is in $\langle \Gamma \rangle$ if and only if R is the universe of a subalgebra of $(A, \text{Pol}(\Gamma))^n$.*

3 Classification tools

In this section, we establish a number of conditions under which the expressivity of a set of operations can be deduced.

Definition 3.1 *An operation $f : A^k \to A$ is a-*implicative *(for $a \in A$) if there exists $i \in [k]$ such that for all $\overline{t} \in A^k$, if $t_i = a$, then $f(\overline{t}) = a$. A set of operations $F \subseteq \mathcal{O}_A$ is a-implicative (for $a \in A$) if every operation in F is a-implicative.*

Theorem 3.2 *Suppose that $F \subseteq \mathcal{O}_A$ is an a-implicative set of operations. Then, F is exponentially expressive.*

Proof. Let $G_n = \{S \subseteq [n] : |S| = \lfloor \frac{n}{2} \rfloor\}$. For every $S \in G_n$, let S' denote the set of n-tuples $\{\overline{t} \in A^n :$ for all $i \in [n], t_i = a \Leftrightarrow i \in S\}$. For every $T \subseteq G_n$, define T' to be the set of n-tuples $\cup_{S \in T} S'$, and define T_F to be the smallest set of n-tuples containing T'. We claim that for every $T \subseteq G_n$, it holds that $T_F \cap (G_n)' = T'$. Observe that when $f : A^k \to A$ is an a-implicative operation with respect to coordinate $i \in [k]$, for all $\overline{t_1}, \ldots, \overline{t_k} \in A^n$, the coordinates at which $f(\overline{t_1}, \ldots, \overline{t_k})$ is equal to a is a superset of the coordinates at which $\overline{t_i}$ is equal to a. Thus, in computing T_F from T', only tuples with $\lfloor \frac{n}{2} \rfloor + 1$ or more a-coordinates are added to T', and $T_F \cap (G_n)' = T'$. It follows that the map taking $T \subseteq G_n$ to T_F is injective, from which we conclude that $\|\mathrm{Inv}(F)\|_n \geq 2^{|G_n|} = 2^{\binom{n}{\lfloor \frac{n}{2} \rfloor}}$. □

As an example application of this theorem, we can show that any semilattice operation is exponentially expressive. Recall that a semilattice operation is a binary operation that is associative, commutative, and idempotent.

Corollary 3.3 *If $f : A^2 \to A$ is a semilattice operation over a finite set A, then $\{f\}$ is exponentially expressive.*

Proof. It is known that for every finite semilattice (A, f), there is an element $m \in A$ such that $f(m, a) = f(a, m) = m$ for all $a \in A$. Hence, f is m-implicative, and exponential expressivity of $\{f\}$ follows from Theorem 3.2. □

Definition 3.4 *An operation $f : A^k \to A$ is* essentially unary *if there exists $i \in [k]$ and a unary operation $g : A \to A$ such that for all $\overline{t} \in A^k$, $f(\overline{t}) = g(t_i)$. A set of operations $F \subseteq \mathcal{O}_A$ is essentially unary if every operation in F is essentially unary.*

Theorem 3.5 *Suppose that $F \subseteq \mathcal{O}_A$ is an essentially unary set of operations. Then, F is exponentially expressive.*

Proof. Let F_u be the set of all unary operations corresponding to the operations in F, and enlargen F_u if necessary so that it is closed under composition. It suffices to show that F_u is exponentially expressive. For all $\overline{a} \in A^n$, define the orbit of $\overline{a}$ to be the set of n-tuples $B_{\overline{a}} = \{f(\overline{a}) : f \in F_u\}$. Select a set $P \subseteq A^n$ so that the set $\{B_{\overline{p}} : \overline{p} \in P\}$ contains all (and only) maximal subsets from the set system $\{B_{\overline{a}} : \overline{a} \in A^n\}$, and such that for all distinct tuples $\overline{p_1}, \overline{p_2} \in P$, $B_{\overline{p_1}} \neq B_{\overline{p_2}}$. In other words, for every maximal orbit S in $\{B_{\overline{a}} : \overline{a} \in A^n\}$, P contains exactly one tuple $\overline{a} \in A^n$ such that $S = B_{\overline{a}}$.

Observe that every tuple $\overline{a} \in A^n$ is contained in some maximal orbit, that is, for every $\overline{a} \in A^n$, there exists $\overline{p} \in P$ such that $\overline{a} \in B_{\overline{p}}$. Since the size of each maximal orbit $B_{\overline{p}}$ is bounded above by $|F_u|$, it follows that $|P| * |F_u| \geq |A|^n$. Now, we claim that for any subset $Q \subseteq P$, the smallest relation $Q' \in \mathrm{Inv}(F_u)$ containing Q has the property that $Q' \cap P = Q$. This is because $Q' = \{f(\overline{q}) : \overline{q} \in Q, f \in F_u\} = \cup\{B_{\overline{q}} : \overline{q} \in Q\}$, so if there was a tuple $\overline{p} \in P$ in $Q' \setminus Q$, there would exist $\overline{q} \in Q$ such that $B_{\overline{p}} \subseteq B_{\overline{q}}$, implying that $B_{\overline{p}}$ would be either non-maximal or equal to $B_{\overline{q}}$ for some $q \in Q$, a contradiction to the definition of P.

Therefore, the mapping taking $Q \subseteq P$ to $Q' \in \mathrm{Inv}(F)$ is injective, and $\|\mathrm{Inv}(F)\|_n$ is bounded below by $2^{|P|}$, which in turn is bounded below by $2^{|A|^n/|F_u|}$. □

Definition 3.6 *An operation $f : A^k \to A$ of rank $k \geq 3$ is a* near unanimity *operation if for all $\overline{t} \in A^k$, if there exists $a \in A$ and $i \in [k]$ such that $t_j = a$ for all $j \in [k] \setminus \{i\}$, then $f(\overline{t}) = a$.*

In other words, $f : A^k \to A$ is a near unanimity operation if for all $x, y \in A$,
$x = f(y, x, x, \ldots, x, x, x) = f(x, y, x, \ldots, x, x, x) = \cdots = f(x, x, x, \ldots, x, y, x)$
$= f(x, x, x, \ldots, x, x, y)$.

Theorem 3.7 *If $F \subseteq O_A$ is a set of operations containing a near unanimity operation, then F is polynomially expressive.*

Proof. By the characterization of relations closed under a near-unanimity operation in [5], every relation in Inv(F) can be expressed by a formula with no existentially quantified variables and containing only relations of arity less than or equal to $k-1$. On n variables, the number of such formulas can be bounded above by $r(n) = \sum_{i=1}^{k} 2^{|A|^i} \binom{n}{i}$, which is polynomial in n. It follows that F is polynomially expressive, since $\|\text{Inv}(F)\|_n \leq 2^{r(n)}$. $\square$

Definition 3.8 *An operation $f : A^3 \to A$ is a* coset operation *if it is of the form $f(a, b, c) = a * (b^{-1}) * c$, where $*$ is a binary operation and ${}^{-1}$ is a unary operation such that $(A, *, {}^{-1})$ is a group.*

Theorem 3.9 *If $F \subseteq \mathcal{O}_A$ is a set of operations containing a coset operation, then F is polynomially expressive.*

Proof. Suppose that R is an arity n relation in Inv(F) It is straightforward to verify that R is a coset of a subgroup of the group $(A, *, {}^{-1})^n$. We show that any subgroup H of $(A, *, {}^{-1})^n$ can be generated by a set of size at most $1 + \log_2 |H|$, from which it follows that any $R \in$ Inv(F) of arity n has a polynomial (in n) size description, and hence that F is polynomially expressive.

If H has no nontrivial proper subgroups, then H is generated by a single element, and the claim is obvious. Otherwise, let J be a maximal proper subgroup of H. The subgroup H is generated by any element of $H \setminus J$ combined with any set of generators for J. By Lagrange's theorem, $|J| \leq |H|/2$ and so by induction J can be generated by a set of size at most $\log_2 |H|$. $\square$

4 Dichotomy Theorem

Theorem 4.1 *For every constraint language $\Gamma \subseteq \mathcal{R}_A$ over a domain A of size two, Γ is either polynomially expressive or exponentially expressive.*

Before proving this theorem, we fix some notation and conventions. Let A be the domain $\{0, 1\}$. Let c_0 and c_1 denote the 0-ary constant operations equal to 0 and 1, respectively. Let $\neg$ denote the usual NOT operation, and let $\vee$, $\wedge$ and $\oplus$ denote the usual binary operations OR, AND, and XOR (over A). Let IMP denote the binary operation defined by IMP$(x, y) = \neg x \vee y$. Let $l : A^3 \to A$ be the ternary operation defined by $l(x, y, z) = x \oplus y \oplus z$, and let $m : A^3 \to A$ be the majority operation over A, that is, the operation defined by $m(x, y, z) = (x \wedge y) \vee (y \wedge z) \vee (x \wedge z)$. Let $h_n : A^{n+1} \to A$ be the operation defined by $h(x_1, \ldots, x_{n+1}) = \vee_{i=1}^{n+1} \wedge_{j \neq i} x_j$.

When $F \subseteq \mathcal{O}_A$ is a set of operations, we use Clo(F) to denote the clone generated by F. For an operation $f : A^k \to A$, let dual(f) denote the dual of f, that is, the operation defined by $(\text{dual}(f))(x_1, \ldots, x_k) = \neg f(\neg x_1, \ldots, \neg x_k)$. The dual of a clone $F \subseteq \mathcal{O}_A$ is the set of operations $\{\text{dual}(f) : f \in F\}$, which is easily verified to be a clone.

Proof. We use Post's classification theorem to demonstrate that every clone over the domain $A = \{0, 1\}$ is either polynomially expressive or exponentially expressive, from which the theorem follows by appeal to Proposition 2.16 and Proposition 2.18. We adopt the notation of [1], to which we refer the reader for a nice description of Post's theorem. Assume that $F \subseteq \mathcal{O}_A$ is a clone; the following cases are exhaustive, up to duality [1].

- Case $F \subseteq V \stackrel{\text{def}}{=} \text{Clo}(\{\vee, c_0, c_1\})$: When $F \subseteq V_2 \stackrel{\text{def}}{=} \text{Clo}(\{\vee\})$, Corollary 3.3 implies the exponential expressivity of F; a similary argument applies to F in the case that $F \subseteq V$.
- Case $F \subseteq N \stackrel{\text{def}}{=} \text{Clo}(\{\text{NOT}, c_1\})$: Every operation in N and hence F is essentially unary, from which the exponential expressivity of F follows from Theorem 3.5.
- Case $F \subseteq S_0 \stackrel{\text{def}}{=} \text{Clo}(\{\text{IMP}\})$: Every operation in S_0 and hence F is 1-implicative, from which the exponential expressivity of F follows from Theorem 3.2.
- Case $F \supseteq S_{00}^n \stackrel{\text{def}}{=} \text{Clo}(\{x \vee (y \wedge z), \text{dual}(h_n)\})$: The operation $\text{dual}(h_n)$ is near-unanimity, from which the polynomial expressivity of F follows from Theorem 3.7.
- Case $F \supseteq L_2 \stackrel{\text{def}}{=} \text{Clo}(\{l\})$: The operation l is a coset operation, from which the polynomial expressivity of F follows from Theorem 3.9.
- Case $F \supseteq D_2 \stackrel{\text{def}}{=} \text{Clo}(\{m\})$: The operation m is near-unanimity, from which the polynomial expressivity of F follows from Theorem 3.7.

Note that throughout, we have implicitly used the fact that for $A, B \subseteq \mathcal{O}_A$, $A \subseteq B$ implies that $\|\text{Inv}(B)\|_n \leq \|\text{Inv}(A)\|_n$. $\square$

It can be deduced from the proof of Theorem 4.1 in conjunction with Schaefer's theorem that for every finite constraint language Γ over a two-element domain, if Γ is polynomially expressive, then the problem CSP(Γ) is tractable, that is, decidable in polynomial time. We conjecture that this phenomenon occurs also in larger domains, that is, for all constraint languages over a finite domain, polynomial expressivity implies tractability.

References

[1] E. Böhler and H. Vollmer. Boolean functions and post's lattice with applications to complexity theory. Lecture Notes for Logic and Interaction 2002.

[2] Marco Cadoli and Francesco M. Donini. A survey on knowledge compilation. *AI Communications*, 10(3-4):137–150, 1997.

[3] Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity Classification of Boolean Constraint Satisfaction Problems*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 2001.

[4] Peter Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.

[5] Peter Jeavons, David Cohen, and Martin Cooper. Constraints, consistency, and closure. *Artificial Intelligence*, 101(1-2):251–265, 1998.

[6] Peter Jeavons, David Cohen, and Justin Pearson. Constraints and universal algebra. *Annals of Mathematics and Artificial Intelligence*, 24(1-4):51–67, 1998.

[7] P.G.Jeavons, D.A.Cohen, and M.Gyssens. Closure properties of constraints. *Journal of the ACM*, 44:527–548, 1997.

[8] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 216–226, 1978.