Computability of Algebraic and Definable Closure

Nathanael Ackerman[1], Cameron Freer[2], and Rehana Patel[3]

[1] Harvard University, Cambridge, MA 02138, USA
nate@aleph0.net
[2] Massachusetts Institute of Technology, Cambridge, MA 02139, USA
freer@mit.edu
[3] African Institute for Mathematical Sciences, M'bour–Thiès, Senegal
rpatel@aims-senegal.org

Abstract. We consider computability-theoretic aspects of the algebraic and definable closure operations for formulas. We show that for φ a Boolean combination of Σ_n-formulas and in a given computable structure, the set of parameters for which the closure of φ is finite is Σ^0_{n+2}, and the set of parameters for which the closure is a singleton is Δ^0_{n+2}. In addition, we construct examples witnessing that these bounds are tight.

Keywords: algebraic closure, definable closure, computable model theory

1 Introduction

An important step towards understanding the relationship between model theory and computability theory is to calibrate the effective content of concepts that are fundamental in classical model theory. There is a long history of efforts to understand this calibration within computable model theory; see, e.g., [Har98].

In this paper, we study the computability of two particular model-theoretic concepts, namely the related notions of *algebraic closure* and *definable closure*, which provide natural characterizations of a "neighborhood" of a set; for more details, see [Hod93, §4.1]. In recent years, the property of a structure having *trivial* definable closure (i.e., the definable closure of every finite set is itself), or equivalently, trivial algebraic closure, has played an important role in combinatorial model theory and descriptive set theory; for some characterizations in terms of this property see, e.g., [CSS99], [AFP16], and [CK18].

The standard notions of algebraic and definable closure can be refined by carrying out a formula-by-formula analysis. We consider the computational strength of the problem of identifying the algebraic or definable closure of a *formula* in a computable structure, and we give tight bounds on the complexity of both. Further, when the formula is quantifier-free, we achieve tightness of these bounds via structures that are model-theoretically "nice", namely, are $\aleph_0$-categorical or of finite Morley rank.

October 28, 2019

1.1 Preliminaries

For standard notions from computability theory, see, e.g., [Soa16]. We write $\{e\}(n)$ to represent the output of the eth Turing machine run on input n, if it converges, and in this case write $\{e\}(n)\downarrow$. Define $W_e := \{n \in \mathbb{N} : \{e\}(n)\downarrow\}$ and $\mathrm{Fin} := \{e \in \mathbb{N} : W_e \text{ is finite}\}$. Recall that Fin is Σ^0_2-complete [Soa16, Theorem 4.3.2]).

In this paper we will focus on computable languages that are relational. Note that this leads to no loss of generality due to the standard fact that computable languages with function or constant symbols can be interpreted computably in relational languages where there is a relation for the graph of each function. For the definitions of languages, first-order formulas, and structures, see [Hod93].

We will work with many-sorted languages and structures; for more details, see [TZ12, §1.1]. Let $\mathcal{L}$ be a (many-sorted) language, let $\mathcal{A}$ be an $\mathcal{L}$-structure, and suppose that $\overline{a}$ is a tuple of elements of $\mathcal{A}$. We say that the **type** of $\overline{a}$ is $\prod_{i\leq n} X_i$ when $\overline{a} \in \prod_{i\leq n}(X_i)^{\mathcal{A}}$, where each of $X_0, \ldots, X_{n-1}$ is a sort of $\mathcal{L}$. The type of a tuple of variables is the product of the sorts of its constituent variables (in order). The type of a relation symbol is defined to be the type of the tuple of its free variables, and similarly for formulas. We write $(\forall \overline{x} : X)$ and $(\exists \overline{x} : X)$ to quantify over a tuple of variables $\overline{x}$ of type X (which includes the special case of a single variable of a given sort).

If we so desired, we could encode each sort using a unary relation symbol, and this would not affect most of our results. However, in Section 3 we are interested in how model-theoretically complicated the structures we build are, and if we do not allow sorts then the construction in Proposition 9 providing a lower bound on the complexity of algebraic closure will not yield an $\aleph_0$-categorical structure.

We now define computable languages and structures.

Definition 1. *Suppose* $\mathcal{L} = \big((X_j)_{j\in J},\ (R_i)_{i\in I}\big)$ *is a language, where* $I, J \in \mathbb{N} \cup \{\mathbb{N}\}$ *and* $(X_j)_{j\in J}$ *and* $(R_i)_{i\in I}$ *are collections of sorts and relation symbols, respectively. Let* $\mathrm{ty}_{\mathcal{L}} \colon I \to J^{<\omega}$ *be such that for all* $i \in I$*, we have* $\mathrm{ty}_{\mathcal{L}}(i) = (j_0, \ldots, j_{n-1})$ *where the type of* R_i *is* $\prod_{k<n} X_{j_k}$*. We say that* $\mathcal{L}$ *is a* **computable language** *when* $\mathrm{ty}_{\mathcal{L}}$ *is a computable function. For each computable language, we fix a computable encoding of all first-order formulas of the language.*

A **computable $\mathcal{L}$-structure** $\mathcal{A}$ *is an* $\mathcal{L}$*-structure with computable underlying set such that the sets* $\{(a, j) : a \in X_j^{\mathcal{A}}\}$ *and* $\{(\overline{b}, i) : \overline{b} \in R_i^{\mathcal{A}}\}$ *are computable subsets of the appropriate domains.*

We say that $c \in \mathbb{N}$ *is a* **code for a structure** *if* $\{c\}(0)$ *is a code for a computable language (via some fixed enumeration of functions of the form* $\mathrm{ty}_{\mathcal{L}}$*) and* $\{c\}(1)$ *is a code for some computable structure in that language. In this case, we write* $\mathcal{L}_c$ *for the language that* $\{c\}(0)$ *codes,* $\mathcal{M}_c$ *for the structure that* $\{c\}(1)$ *codes, and* T_c *for the first-order theory of* $\mathcal{M}_c$*. Let* CompStr *be the collection of* $c \in \mathbb{N}$ *that are codes for structures.*

Note that these notions relativize in the obvious way. For more details on basic notions in computable model theory, see [Har98].

Towards defining algebraic closure and definable closure for formulas, we first describe when a formula is algebraic or definable at a given tuple.

Definition 2. *Let $\varphi(\overline{x};\overline{y})$ be a first-order $\mathcal{L}$-formula, let $\mathcal{A}$ be an $\mathcal{L}$-structure, and suppose $\overline{a} \in \mathcal{A}$ has the same type as $\overline{x}$.*

- *The formula $\varphi(\overline{x};\overline{y})$ is* **algebraic at** *$\overline{a}$ if*
$$\mathrm{cl}_{\varphi,\mathcal{A}}(\overline{a}) := \{\overline{b} \in \mathcal{A} \,:\, \mathcal{A} \models \varphi(\overline{a};\overline{b})\}$$
is finite (possibly empty).
- *The formula $\varphi(\overline{x};\overline{y})$ is* **definable at** *$\overline{a}$ if $\big|\mathrm{cl}_{\varphi,\mathcal{A}}(\overline{a})\big| = 1$.*

We now describe several sets that encode those formulas that are algebraic or definable at given tuples. These are our analogues, for individual formulas, of the standard notions of algebraic closure and definable closure. See [Hod93, §4.1] for more details on these standard notions.

Definition 3.
- $\mathrm{CL} := \big\{(c, \varphi(\overline{x};\overline{y}), \overline{a}, k) \,:\, c \in \mathrm{CompStr}$, *$\varphi(\overline{x};\overline{y})$ a first-order $\mathcal{L}_c$-formula, $\overline{a} \in \mathcal{M}_c$ having the same type as $\overline{x}$, and $k \in \mathbb{N} \cup \{\infty\}$ with $|\,\mathrm{cl}_{\varphi,\mathcal{M}_c}(\overline{a})| = k\big\}$.*
- $\mathrm{ACL} := \big\{(c, \varphi(\overline{x};\overline{y}), \overline{a}) \,:\,$ *there exists $k \in \mathbb{N}$ with $(c, \varphi(\overline{x};\overline{y}), \overline{a}, k) \in \mathrm{CL}\big\}$.*
- $\mathrm{DCL} := \big\{(c, \varphi(\overline{x};\overline{y}), \overline{a}) \,:\, (c, \varphi(\overline{x};\overline{y}), \overline{a}, 1) \in \mathrm{CL}\big\}$.
- *For $Y \in \{\mathrm{CL}, \mathrm{ACL}, \mathrm{DCL}\}$ and $n \in \mathbb{N}$ let*
$$Y_n := \{t \in Y \,:\, \textit{the second coordinate of } t \textit{ is a Boolean combination of } \Sigma_n\textit{-formulas}\}.$$
- *For $Y \in \{\mathrm{CL}, \mathrm{ACL}, \mathrm{DCL}\} \cup \{\mathrm{CL}_n, \mathrm{ACL}_n, \mathrm{DCL}_n\}_{n\in\mathbb{N}}$ and $c \in \mathrm{CompStr}$, let $Y^c := \{u \,:\, (c)^\wedge u \in Y\}$, i.e., select those elements of Y whose first coordinate is c, and then remove this first coordinate.*

Note that CompStr is a Π^0_2 class. Hence even before we consider the complexity of whether formulas are algebraic or definable at various tuples, the sets $\mathrm{CL}, \mathrm{ACL}, \mathrm{DCL}$ are already complicated computability-theoretically. As such, we will mainly be interested in the question of how complex $\mathrm{CL}^c, \mathrm{ACL}^c, \mathrm{DCL}^c$ can be, when c is a code for a structure. The next three lemmas connect these sets.

Lemma 4. *Uniformly in the parameters $c \in \mathrm{CompStr}$ and $n \in \mathbb{N}$, the set*
$$\{(\varphi(\overline{x};\overline{y}), \overline{a}, k) \in \mathrm{CL}^c_n \,:\, k \in \mathbb{N},\ k \geq 1\}$$
is computably enumerable from DCL^c_n.

Proof. Suppose $\varphi(\overline{x};\overline{y})$ is a Boolean combination of Σ_n-formulas, and let $k \geq 1$. For each $j < k$, choose a tuple of new variables $\overline{z}^{\,j}$ of the same type as $\overline{y}$. Define the formula

$$\Phi_{\varphi(\overline{x};\overline{y}),k} := \bigwedge_{k_0<k_1<k} (\overline{z}^{\,k_0} \neq \overline{z}^{\,k_1}) \wedge \bigwedge_{k_1<k} \varphi(\overline{x};\overline{z}^{\,k_1})$$

which specifies k-many distinct realizations of the tuple $\overline{y}$ in $\varphi(\overline{x};\overline{y})$, given an instantiation of $\overline{x}$. Note that $\Phi_{\varphi(\overline{x};\overline{y}),k}$ is also a Boolean combination of Σ_n-formulas.

For $j < k$, let $\tau_j := \overline{x}\,\overline{z}^{\,0} \cdots \overline{z}^{\,j-1}\,\overline{z}^{\,j+1} \cdots \overline{z}^{\,k-1}$, and write $\Phi_{\varphi(\overline{x};\overline{y}),k}(\tau_j;\overline{z}^{\,j})$ to mean $\Phi_{\varphi(\overline{x};\overline{y}),k}$ considered as a formula whose free variables are partitioned as $(\tau_j,\overline{z}^{\,j})$. Note that $(\varphi(\overline{x};\overline{y}),\overline{a},k) \in \mathrm{CL}_n^c$ if and only if

$$\Big(\Phi_{\varphi(\overline{x};\overline{y}),k}(\tau_j;\overline{z}^{\,j}), \overline{a}\,\overline{b}^{\,0} \cdots \overline{b}^{\,j-1}\,\overline{b}^{\,j+1} \cdots \overline{b}^{\,k-1}\Big) \in \mathrm{DCL}_n^c$$

for some $j < k$ and $\overline{b}^{\,0},\dots,\overline{b}^{\,j-1},\overline{b}^{\,j+1},\dots,\overline{b}^{\,k-1} \in \mathcal{M}_c$. By enumerating over all such parameters, and enumerating over all choices of φ and k, we see that the desired set is c.e. from DCL_n^c. ◻

Lemma 5. *Uniformly in the parameters* $c \in \mathrm{CompStr}$ *and* $n \in \mathbb{N}$*, the set*

$$\big\{\big(\varphi(\overline{x};\overline{y}),\overline{a},k\big) \in \mathrm{CL}_n^c \,:\, k = 0\big\}$$

is computably enumerable from DCL_n^c.

Proof. Suppose $\varphi(\overline{x};\overline{y})$ is a Boolean combination of Σ_n-formulas. Let $\overline{z}$ be a tuple of variables having the same type as $\overline{y}$ and disjoint from $\overline{x}\,\overline{y}$. Let

$$\Psi_{\varphi(\overline{x};\overline{y})}(\overline{x}\,\overline{z};\overline{y}) := \varphi(\overline{x};\overline{y}) \vee (\overline{y} = \overline{z}).$$

Note that $\Psi_{\varphi(\overline{x};\overline{y})}(\overline{x}\,\overline{z};\overline{y})$ is also a Boolean combination of Σ_n-formulas.

Now suppose $\overline{b_0}$ and $\overline{b_1}$ are distinct tuples of elements of $\mathcal{M}_c$ having the same type as $\overline{z}$. Then the following are equivalent:

- $\big(\Psi_{\varphi(\overline{x};\overline{y})}(\overline{x}\,\overline{z};\overline{y}), \overline{a}\,\overline{b_0}\big) \in \mathrm{DCL}_n^c$ and $\big(\Psi_{\varphi(\overline{x};\overline{y})}(\overline{x}\,\overline{z};\overline{y}), \overline{a}\,\overline{b_1}\big) \in \mathrm{DCL}_n^c$;
- $\big\{\overline{b} \,:\, \mathcal{M}_c \models \varphi(\overline{a};\overline{b})\big\} = \emptyset$, i.e., $(\varphi(\overline{x};\overline{y}),\overline{a},0) \in \mathrm{CL}_n^c$.

The result is then immediate. ◻

Lemma 6. *Uniformly in the parameters* $c \in \mathrm{CompStr}$ *and* $n \in \mathbb{N}$*, there are computable reductions in both directions between* $\mathrm{ACL}_n^c \coprod \mathrm{DCL}_n^c$ *and* CL_n^c.

Proof. It is immediate from the definitions that DCL_n^c is computable from CL_n^c. Further, ACL_n^c is computable from CL_n^c as

$$\mathrm{ACL}_n^c = \big\{(\varphi(\overline{x};\overline{y}),\overline{a}) \,:\, (\exists k)\ (\varphi(\overline{x};\overline{y}),\overline{a},k) \in \mathrm{CL}_n^c \text{ and } k \neq \infty\big\}$$

and $(\varphi(\overline{x};\overline{y}),\overline{a},k) \in \mathrm{CL}_n^c$ holds for a unique $k \in \mathbb{N} \cup \{\infty\}$.

Lemmas 4 and 5 together tell us that $\{(\varphi(\overline{x};\overline{y}),\overline{a},k) \in \mathrm{CL}_n^c : k \in \mathbb{N}\}$ is computably enumerable from DCL_n^c. But $(\varphi(\overline{x};\overline{y}),\overline{a},\infty) \in \mathrm{CL}_n^c$ if and only if $(\varphi(\overline{x};\overline{y}),\overline{a}) \notin \mathrm{ACL}_n^c$. Therefore when $\varphi(\overline{x};\overline{y})$ is a Boolean combination of Σ_n-formulas, and given $\overline{a} \in \mathcal{M}_c$, we can compute from ACL_n^c whether or not $(\varphi(\overline{x};\overline{y}),\overline{a},\infty) \in \mathrm{CL}_n^c$. Further, if $(\varphi(\overline{x};\overline{y}),\overline{a},\infty) \notin \mathrm{CL}_n^c$, then we can compute from DCL_n^c the (unique) value of k such that $(\varphi(\overline{x};\overline{y}),\overline{a},k) \in \mathrm{CL}_n^c$. Hence CL_n^c is computable from the set $\mathrm{ACL}_n^c \coprod \mathrm{DCL}_n^c$. □

Note that by Lemma 6 we are justified, from a computability-theoretic perspective, in restricting our attention to ACL and DCL (and their variants), as opposed to CL.

2 Upper Bounds for Quantifier-Free Formulas

We now provide straightforward upper bounds on the complexity of ACL_0^c and DCL_0^c for $c \in \mathrm{CompStr}$.

Proposition 7. *Uniformly in the parameter $c \in \mathrm{CompStr}$, the set ACL_0^c is a Σ_2^0 class.*

Proof. Uniformly in $c \in \mathrm{CompStr}$, a quantifier-free $\mathcal{L}_c$-formula $\varphi(\overline{x};\overline{y})$, and tuple $\overline{a} \in \mathcal{M}_c$ of the same type as $\overline{x}$, we can computably find an $e \in \mathbb{N}$ such that W_e equals $\mathrm{cl}_{\varphi,\mathcal{M}_c}(\overline{a})$ (where the elements of $\mathrm{cl}_{\varphi,\mathcal{M}_c}(\overline{a})$ are encoded in $\mathbb{N}$ in a standard way).

Further, $(\varphi(\overline{x};\overline{y}),\overline{a}) \in \mathrm{ACL}_0^c$ if and only if $\mathrm{cl}_{\varphi,\mathcal{M}_c}(\overline{a})$ is finite. Therefore ACL_0^c is Σ_2^0 as Fin is Σ_2^0. □

Proposition 8. *Uniformly in the parameter $c \in \mathrm{CompStr}$, the set DCL_0^c is the intersection of a Π_1^0 and a Σ_1^0 class (in particular, it is a Δ_2^0 class).*

Proof. Uniformly in $c \in \mathrm{CompStr}$, the set of all tuples $(\varphi(\overline{x};\overline{y}),\overline{a})$ such that

$$\mathcal{M}_c \models (\forall \overline{y_0},\overline{y_1})\ \big((\varphi(\overline{a};\overline{y_0}) \wedge \varphi(\overline{a};\overline{y_1})) \rightarrow (\overline{y_0} = \overline{y_1})\big)$$

holds is a Π_1^0 class. Likewise, uniformly in $c \in \mathrm{CompStr}$, the set of all tuples $(\varphi(\overline{x};\overline{y}),\overline{a})$ such that there exists $\overline{b}$ with $\mathcal{M}_c \models \varphi(\overline{a};\overline{b})$ is a Σ_1^0 class. □

As a consequence, DCL_0^c is computable from $\mathbf{0}'$.

3 Lower Bounds for Quantifier-Free Formulas

We now show that the upper bounds in Section 2 are tight. Further, we do so using structures that have nice model-theoretic properties.

We first show that the upper bound in Proposition 7 is tight.

Proposition 9. *There is a parameter $c \in \mathrm{CompStr}$ such that the following hold.*

(a) *$\mathcal{L}_c$ has no relation symbols, i.e., $\mathcal{L}_c$ consists only of sorts.*
(b) *For each ordinal α, the theory T_c has $(|\alpha + 1|^{\omega})$-many models of size $\aleph_\alpha$. In particular, T_c is $\aleph_0$-categorical.*
(c) $\mathrm{ACL}_0^c \equiv_1 \mathrm{Fin}$. *In particular,* ACL_0^c *is a Σ_2^0-complete set.*

Proof. Let $\big((e_i, n_i)\big)_{i\in\mathbb{N}}$ be a computable enumeration without repetition of

$$\{(e,n) \,:\, e,n \in \mathbb{N} \text{ and } \{e\}(n)\!\downarrow\}.$$

Note that for each $\ell \in \mathbb{N} \cup \{\infty\}$, there are infinitely many programs that halt on exactly ℓ-many inputs, and so there are infinitely many $e \in \mathbb{N}$ that are equal to e_i for exactly ℓ-many i.

Let $c \in \mathrm{CompStr}$ be such that

- $\mathcal{L}_c$ consists of infinitely many sorts $(X_i)_{i\in\mathbb{N}}$ and no relation symbols,
- the underlying set of $\mathcal{M}_c$ is $\mathbb{N}$, and
- for each $i \in \mathbb{N}$, the element i is of sort X_{e_i} in $\mathcal{M}_c$.

A model of T_c is determined up to isomorphism by the number of elements in the instantiation of each sort. Hence there are $\aleph_0$-many sorts of each finite size and $\aleph_0$-many that are infinite (each of which may have size $\aleph_\beta$ for arbitrary $\beta \le \alpha$, in a model of size $\aleph_\alpha$), and so (b) holds.

Now $|W_e| = |(X_e)^{\mathcal{M}_c}|$ and so Fin is 1-equivalent to $\{e \,:\, (X_e)^{\mathcal{M}_c} \text{ is finite}\}$. Recall that each variable in a many-sorted language is assigned a single sort, and so no non-trivial Boolean combination of instantiations of sorts is definable. Since there are no relation symbols in $\mathcal{L}_c$, every quantifier-free definable set is contained in some product of instantiations of sorts, and is itself the product of finite or cofinite subsets of instantiations of sorts. Therefore ACL_0^c is 1-equivalent to $\{e \,:\, (X_e)^{\mathcal{M}_c} \text{ is finite}\}$ as well, establishing (c). □

We now show that the upper bound in Proposition 8 is tight.

Proposition 10. *There is a parameter $c \in \mathrm{CompStr}$ such that the following hold.*

(a) *The language $\mathcal{L}_c$ has one sort and a single binary relation symbol E.*
(b) *The structure $\mathcal{M}_c$ is a countable saturated model of T_c with underlying set $\mathbb{N}$.*
(c) *For each ordinal α, the theory T_c has $(|\alpha + \omega|)$-many models of size $\aleph_\alpha$, and has finite Morley rank.*
(d) *There is a computable array $\big(U_{k,\ell}\big)_{k,\ell\in\mathbb{N}}$ of subsets of $\mathbb{N}$ such that every countable model of T_c is isomorphic to the restriction of $\mathcal{M}_c$ to underlying set $U_{k,\ell}$ for exactly one pair (k, ℓ).*
(e) *If $\mathcal{N} \cong \mathcal{M}_c$ then uniformly in $\mathcal{N}$ we can compute $\mathbf{0}'$ from*

$$\big\{a \,:\, \big|\{b \,:\, \mathcal{N} \models E(a;b)\}\big| = 1\big\}.$$

(f) The set

$$\{a \,:\, (E(x;y), a) \in \mathrm{DCL}_0^c\}$$

has Turing degree $\mathbf{0}'$.

Proof. Let $g\colon \mathbb{N} \to \{0,1\}$ be the characteristic function of $\mathbf{0}'$, i.e., such that $g(n) = 1$ if and only if $n \in \mathbf{0}'$. As $\mathbf{0}'$ is a Δ^0_2 set, there is some computable function $f\colon \mathbb{N}\times\mathbb{N} \to \{0,1\}$ such that $\lim_{s\to\infty} f(n,s) = g(n)$ for all $n \in \mathbb{N}$.

We will construct $\mathcal{M}_c$ in the language specified in (a) so as to satisfy the following axioms.

- $(\forall x)\ \neg E(x,x)$
- $(\forall x,y)\ (E(x,y) \to E(y,x))$
- $(\forall x)(\exists y)\ E(x,y)$
- $(\forall x)(\exists^{\le 2} y)\ E(x,y)$

By a *graph* we mean a structure with a single undirected irreflexive binary relation. A *chain* in a graph is a connected component of the graph each of whose vertices has degree 1 or 2; hence a chain either is finite with at least two vertices, or is infinite on one side (an $\mathbb{N}$-chain), or is infinite on both sides (a $\mathbb{Z}$-chain). By the *order* of a chain we mean its number of vertices.

The above axioms specify that $\mathcal{M}_c$ will be a graph (with edge relation E) that is the union of chains. In fact, we will construct $\mathcal{M}_c$ so as to have infinitely many chains of certain finite orders, infinitely many $\mathbb{N}$-chains, and infinitely many $\mathbb{Z}$-chains.

For $n \in \mathbb{N}$, let p_n denote the nth prime number. We now construct $\mathcal{M}_c$ with underlying set $\mathbb{N}$, in stages.

Stage 0:
Let $\{N_i\}_{i\in\mathbb{N}} \cup \{Z_i\}_{i\in\mathbb{N}} \cup \{F\}$ be a uniformly computable partition of $\mathbb{N}$ into infinite sets.

For each $i \in \mathbb{N}$, let the induced subgraph on N_i be an $\mathbb{N}$-chain, and let the induced subgraph on Z_i be a $\mathbb{Z}$-chain. The only other edges will be between elements of F (to be determined in later stages).

Stage $2s+1$:
Let a_s be the least element of F that is not yet part of an edge. Create a finite chain of order $(p_s)^{2+f(s,s)}$ consisting of a_s and other elements of F not yet in any edge.

Stage $2s+2$:
For each $n \le s$, we have two cases, based on the values of f. If $f(n,s) = f(n,s+1)$, do nothing.

Otherwise, if $f(n,s) \ne f(n,s+1)$, consider the (unique) chain whose order so far is $(p_n)^k$ for some positive k. Extend this chain by $\big((p_n)^{k+1} - (p_n)^k\big)$-many elements of

F which are not yet in any edge, so that the resulting chain has order $(p_n)^{2\ell+f(n,s+1)}$ for some $\ell \in \mathbb{N}$.

The resulting graph is computable, as every vertex participates in at least one edge, and whether or not there is an edge between a given pair of vertices is determined by the first stage at which each vertex of the pair becomes part of some edge.

Observe that every element of F is part of a chain of elements of F whose order is some positive power of a prime, which moreover is the only chain in $\mathcal{M}_c$ whose order is a power of that prime.

Now, every model of T_c is determined by the number of $\mathbb{N}$-chains and the number of $\mathbb{Z}$-chains in it. In a model of size $\aleph_\alpha$, there must be either $\aleph_\alpha$-many $\mathbb{N}$-chains and 0-, 1-, …, $\aleph_0$-, …, or $\aleph_\alpha$-many $\mathbb{Z}$-chains, or vice-versa. Condition (b) holds because the countable saturated models of T_c have $\aleph_0$-many $\mathbb{N}$-chains and $\aleph_0$-many $\mathbb{Z}$-chains, as does $\mathcal{M}_c$. Condition (c) holds because none of these $\mathbb{N}$-chains or $\mathbb{Z}$-chains are first-order definable.

For condition (d), let $U_{k,\ell} := \bigcup_{i<k} N_i \cup \bigcup_{i<\ell} Z_i \cup F$.

Towards condition (e), note that for each $n \in \mathbb{N}$, there is a unique chain of order a power of p_n. Writing $(p_n)^{j_n}$ for this order, we have $j_n \equiv g(n) \pmod 2$. An element $a \in \mathcal{N}$ is one of the two ends of a finite chain or the beginning of an $\mathbb{N}$-chain if and only if $|\{b : \mathcal{N} \models E(a;b)\}| = 1$. So, from the set $\{a : |\{b : \mathcal{N} \models E(a;b)\}| = 1\}$ we can enumerate the orders of all finite chains, and hence can compute $g(n)$ for all n.

Finally, recall that DCL_0^c is computable from $\mathbf{0}'$ and so $\{a : (E(x;y), a) \in \mathrm{DCL}_0^c\}$ is also computable from $\mathbf{0}'$. Hence (f) follows from (e). $\square$

4 Boolean Combinations of Σ_n-Formulas

We now study the complexity of ACL^c and DCL^c with respect to Boolean combinations of Σ_n-formulas.

The following lemma captures a computable version of the standard process known as *Morleyization*. The proof is straightforward.

Lemma 11. *Let $\mathcal{L}$ be a computable language and $\mathcal{A}$ a computable $\mathcal{L}$-structure. For each $n \in \mathbb{N}$ there is a computable language $\mathcal{L}_n$ and a $\mathbf{0}^{(n)}$-computable $\mathcal{L}_n$-structure $\mathcal{A}_n$ such that*

- $\mathcal{L} \subseteq \mathcal{L}_n \subseteq \mathcal{L}_{n+1}$,
- *$\mathcal{A}$ is the reduct of $\mathcal{A}_n$ to the language $\mathcal{L}$,*
- *for each first-order $\mathcal{L}_n$-formula φ there is a first-order $\mathcal{L}$-formula ψ_φ (of the same type as φ) such that*

$$\mathcal{A}_n \models (\forall x_0, \ldots, x_{k-1})\ \varphi(x_0, \ldots, x_{k-1}) \leftrightarrow \psi_\varphi(x_0, \ldots, x_{k-1}),$$

where k is the number of free variables of φ, and

- *for each first-order $\mathcal{L}$-formula ψ, if ψ is a Boolean combination of Σ_n-formulas then there is a first-order quantifier-free $\mathcal{L}_n$-formula φ_ψ (of the same type as ψ) such that*

$$\mathcal{A}_n \models (\forall x_0, \ldots, x_{k-1})\ \psi(x_0, \ldots, x_{k-1}) \leftrightarrow \varphi_\psi(x_0, \ldots, x_{k-1}),$$

where k is the number of free variables of ψ.

Lemma 11 tells us that the methods used earlier in this paper to study quantifier-free algebraic and definable closures can be applied to more complicated formulas, provided that we allow the structures that we build to have greater complexity, as we now illustrate.

Corollary 12. *For every $n \in \mathbb{N}$ and $c \in \mathrm{CompStr}$,*

- ACL_n^c *is a Σ_{n+2}^0 class, and*
- DCL_n^c *is a Δ_{n+2}^0 class.*

Proof. By Lemma 11, we know that ACL_n is equivalent to the relativization of ACL_0 to the class of structures computable in $\mathbf{0}^{(n)}$, and that DCL_n is equivalent to the relativization of DCL_0 to the class of structures computable in $\mathbf{0}^{(n)}$.

Therefore by Propositions 7 and 8, ACL_n^c is a $\Sigma_2^0(\mathbf{0}^{(n)})$ class and DCL_n^c is a $\Delta_2^0(\mathbf{0}^{(n)})$ class. □

In Theorem 15 we will show that these bounds are tight. Towards this, we will need the next two results.

Suppose that $\mathcal{L}$ is a language containing a sort N and a relation symbol S of type $N \times N$. Let $\mathcal{A}$ be an $\mathcal{L}$-structure. We call $(N^\mathcal{A}, S^\mathcal{A})$ a **directed $\mathbb{N}$-chain** when it is isomorphic to a single-sorted structure with underlying set $\mathbb{N}$ in a language consisting of the binary relation symbol S, in which $S(k, \ell)$ holds precisely when $\ell = k + 1$. In other words, $(N^\mathcal{A}, S^\mathcal{A})$ is a directed $\mathbb{N}$-chain if there is an isomorphism between it and $\mathbb{N}$ with its successor function viewed as a directed graph. Note that this isomorphism is necessarily unique. Given $\ell \in \mathbb{N}$, we write $\widehat{\ell}$ to denote the corresponding element of $N^\mathcal{A}$ according to this isomorphism.

Lemma 13. *Let $\mathcal{L}$ be a language containing a sort N and a relation symbol S of type $N \times N$ (and possibly other sorts and relation symbols). Let $\mathcal{A}$ be an $\mathcal{L}$-structure such that $(N^\mathcal{A}, S^\mathcal{A})$ is a directed $\mathbb{N}$-chain. Let $k \in \mathbb{N}$ and let $h(\overline{x}, m)$ be an $\mathcal{L}$-formula that is a Boolean combination of Σ_k-formulas, where $\overline{x}$ is of some type X, and m has sort N.*

Suppose that

$$\mathcal{A} \models (\forall \overline{x} : X)(\exists^{\leq 1} m : N)(\exists p : N)\ S(m, p) \wedge \big(h(\overline{x}, m) \leftrightarrow \neg h(\overline{x}, p)\big).$$

Let $H \colon X^\mathcal{A} \times \mathbb{N} \to \{\mathrm{True}, \mathrm{False}\}$ be the function where $H(\overline{a}, \ell) = \mathrm{True}$ if and only if $\mathcal{A} \models h(\overline{a}, \widehat{\ell})$. Note that $\lim_{\ell \to \infty} H(\overline{a}, \ell)$ exists for all $\overline{a} \in X^\mathcal{A}$.

There is an $\mathcal{L}$-formula $h'(\overline{x})$, where $\overline{x}$ is of type X, such that h' is a Boolean combination of Σ_{k+1}-formulas and for all $\overline{a} \in X^{\mathcal{A}}$,

$$\mathcal{A} \models h'(\overline{a}) \quad \textit{if and only if} \quad \lim_{m\to\infty} H(\overline{a}, m) = \text{True}.$$

Proof. Define the formula h' by

$$h'(\overline{x}) := \big[(\forall m : N)\ h(\overline{x}, m)\big] \quad \vee \quad \big[(\exists m, p : N)\ \big(\neg h(\overline{x}, m) \wedge h(\overline{x}, p) \wedge S(m, p)\big)\big].$$

Clearly h' is a Boolean combination of Σ_{k+1}-formulas with the desired property. □

Proposition 14. *Let $n \in \mathbb{N}$ and let $\mathcal{L}$ be a language containing a sort N and a relation symbol S of type $N \times N$ (and possibly other sorts and relation symbols). Suppose $\mathcal{A}$ is an $\mathcal{L}$-structure that is computable in $\mathbf{0}^{(n)}$ and such that $(N^{\mathcal{A}}, S^{\mathcal{A}})$ is a computable directed $\mathbb{N}$-chain. Then there is a computable language $\mathcal{L}^+$ and a computable $\mathcal{L}^+$-structure $\mathcal{A}^+$ such that for every relation symbol $R \in \mathcal{L}$ other than S, there is an $\mathcal{L}^+$-formula φ_R that is a Boolean combination of Σ_n-formulas for which $R^{\mathcal{A}} = (\varphi_R)^{\mathcal{A}^+}$.*

Proof. We begin by defining, for relation symbols in $\mathcal{L}$ other than S, certain auxiliary functions. Let R be a relation symbol in $\mathcal{L}$ that is not S, and let X be its type. For every $k \in \mathbb{N}$ such that $0 \le k \le n$, there is some $\mathbf{0}^{(n-k)}$-computable function $F_{R,k} \colon X^{\mathcal{A}} \times \mathbb{N}^k \to \{\text{True}, \text{False}\}$ such that for all $\overline{a} \in X^{\mathcal{A}}$, the following hold:

- $F_{R,0}(\overline{a}) = 1$ if and only if $\mathcal{A} \models R(\overline{a})$.
- Suppose $k \ge 1$ and let $\ell_0, \dots, \ell_{k-2} \in \mathbb{N}$. There is at most one $s \in \mathbb{N}$ for which

$$F_{R,k}(\overline{a}, \ell_0, \dots, \ell_{k-2}, s) \neq F_{R,k}(\overline{a}, \ell_0, \dots, \ell_{k-2}, s+1).$$

  Further,

$$F_{R,k-1}(\overline{a}, \ell_0, \dots, \ell_{k-2}) = \lim_{\ell_{k-1}\to\infty} F_{R,k}(\overline{a}, \ell_0, \dots, \ell_{k-2}, \ell_{k-1}).$$

Next we define the computable language $\mathcal{L}^+$ as follows:

- $\mathcal{L}^+$ has the same sorts as $\mathcal{L}$.
- For each relation symbol $R \in \mathcal{L}$ other than S, there is a relation symbol $R^+ \in \mathcal{L}^+$ of type $X \times N^n$, where X is the type of R.

  Now define the computable $\mathcal{L}^+$-structure $\mathcal{A}^+$ as follows:

- $\mathcal{A}^+$ has the same underlying set as $\mathcal{A}$, and sorts are instantiated on the same sets in $\mathcal{A}^+$ as in $\mathcal{A}$.
- $S^{\mathcal{A}^+}$ is the same relation as $S^{\mathcal{A}}$.
- For each $R \in \mathcal{L}$ other than S, each tuple $\overline{a} \in X^{\mathcal{A}^+}$ where X is the type of R, and any $\ell_0, \dots, \ell_{n-1} \in \mathbb{N}$, we have

$$\mathcal{A}^+ \models R^+(\overline{a}, \widehat{\ell_0}, \ldots, \widehat{\ell_{n-1}}) \text{ if and only if } F_{R,n}(\overline{a}, \ell_0, \ldots, \ell_{n-1}) = \text{True}.$$

(Recall that for $\ell \in \mathbb{N}$, we have defined $\widehat{\ell} \in N^{\mathcal{A}^+}$ to be the ℓ^{th} element of the directed $\mathbb{N}$-chain.)

Finally, we build, for each relation symbol $R \in \mathcal{L}$ other than S, an $\mathcal{L}^+$-formula φ_R. First apply Lemma 13 (with $k = 0$) to $\mathcal{A}^+$ and the $\mathcal{L}^+$-formula

$$h_0(\overline{x}y_0 \cdots y_{n-2}, y_{n-1}) := R^+(\overline{x}, y_0, \ldots, y_{n-1})$$

(where $\overline{x}$ has type X and each y_i has type N) to obtain an $\mathcal{L}^+$-formula $h_0'(\overline{x}y_0 \cdots y_{n-2})$ that is a Boolean combination of Σ_1-formulas. Next apply Lemma 13 again (with $k = 1$) to $\mathcal{A}^+$ and the $\mathcal{L}^+$-formula

$$h_1(\overline{x}y_0 \cdots y_{n-3}, y_{n-2}) := h_0'(\overline{x}y_0 \cdots y_{n-2})$$

to obtain an $\mathcal{L}^+$-formula $h_1'(\overline{x}y_0 \cdots y_{n-3})$ that is a Boolean combination of Σ_2-formulas. Proceed in this way for $k = 2, \ldots, n-1$, to obtain an $\mathcal{L}^+$-formula $\varphi_R(\overline{x}) := h_{n-1}'(\overline{x})$ that is a Boolean combination of Σ_n-formulas for which $R^{\mathcal{A}} = (\varphi_R)^{\mathcal{A}^+}$. □

Combining this with results from Section 3, we obtain the following.

Theorem 15. *For each $n \in \mathbb{N}$,*

(a) there is an element $a \in \mathrm{CompStr}$ such that ACL_n^a is a $\Sigma_2^0(\mathbf{0}^{(n)})$-complete set, and
(b) there is an element $b \in \mathrm{CompStr}$ such that $\mathrm{DCL}_n^b \equiv_{\mathrm{T}} \mathbf{0}^{(n+1)}$.

Proof. Let $\mathcal{P}$ be the structure constructed in the proof of Proposition 9, relativized to the oracle $\mathbf{0}^{(n)}$, i.e., so that $\mathcal{P}$ is computable from $\mathbf{0}^{(n)}$. Let the structure $\mathcal{P}^*$ be $\mathcal{P}$ augmented with a sort N (instantiated on a new set of elements) along with a relation symbol S of type $N \times N$, such that $(N^{\mathcal{P}^*}, S^{\mathcal{P}^*})$ is a computable directed $\mathbb{N}$-chain. Part (a) then follows by applying Proposition 14 to $\mathcal{P}^*$ to obtain some computable structure, namely $\mathcal{M}_a$ for some $a \in \mathrm{CompStr}$. Then ACL_n^a is a $\Sigma_2^0(\mathbf{0}^{(n)})$-complete set.

Let $\mathcal{Q}$ be the structure constructed in the proof of Proposition 10 relativized to the oracle $\mathbf{0}^{(n)}$, i.e., so that $\mathcal{Q}$ is computable from $\mathbf{0}^{(n)}$. Let the structure $\mathcal{Q}^*$ be obtained from $\mathcal{Q}$ by similarly augmenting it by N and S, so that $(N^{\mathcal{Q}^*}, S^{\mathcal{Q}^*})$ is a new computable directed $\mathbb{N}$-chain. Part (b) then follows by applying Proposition 14 to $\mathcal{Q}^*$ to obtain a computable structure $\mathcal{M}_b$ for some $b \in \mathrm{CompStr}$. Then we have $\mathrm{DCL}_n^b \equiv_{\mathrm{T}} \mathbf{0}^{(n+1)}$. □

Note that the structures constructed in Theorem 15 do not obviously have the nice model-theoretic properties ($\aleph_0$-categoricity or finite Morley rank) that those constructed in Proposition 9 and Proposition 10 do, because the application of Proposition 14 makes their theories more elaborate.

Question 16. Is there some $c \in \mathrm{CompStr}$ such that ACL_n^c is a $\Sigma_2^0(\mathbf{0}^{(n)})$-complete set or $\mathrm{DCL}_n^c \equiv_{\mathrm{T}} \mathbf{0}^{(n+1)}$ and $\mathcal{M}_c$ is nice model-theoretically (e.g., $\aleph_0$-categorical, strongly minimal, stable, etc.)?

References

[AFP16] N. Ackerman, C. Freer, and R. Patel, *Invariant measures concentrated on countable structures*, Forum Math. Sigma **4** (2016), no. e17, 59 pp.

[CK18] R. Chen and A. S. Kechris, *Structurable equivalence relations*, Fund. Math. **242** (2018), no. 2, 109–185.

[CSS99] G. Cherlin, S. Shelah, and N. Shi, *Universal graphs with forbidden subgraphs and algebraic closure*, Adv. in Appl. Math. **22** (1999), no. 4, 454–491.

[Har98] V. S. Harizanov, *Pure computable model theory*, Handbook of recursive mathematics, Vol. 1, Stud. Logic Found. Math., vol. 138, North-Holland, 1998, pp. 3–114.

[Hod93] W. Hodges, *Model theory*, Encyclopedia of Mathematics and its Applications, vol. 42, Cambridge University Press, Cambridge, 1993.

[Soa16] R. I. Soare, *Turing computability*, Theory and Applications of Computability, Springer-Verlag, Berlin, 2016.

[TZ12] K. Tent and M. Ziegler, *A course in model theory*, Lecture Notes in Logic, vol. 40, Cambridge University Press, 2012.