Traversal-Invariant Characterizations of Logarithmic Space

Siddharth Bhaskar[1]
Department of Computer Science, James Madison University, Harrisonburg, VA, USA

Steven Lindell
Department of Computer Science, Haverford College, Haverford, PA, USA

Scott Weinstein
Department of Philosophy, University of Pennsylvania, Philadelphia, PA, USA

Abstract

We give a novel descriptive-complexity theoretic characterization of L and NL computable queries over finite structures using *traversal invariance*. We summarize this as (N)L = FO + (breadth-first) traversal-invariance.

2012 ACM Subject Classification Theory of computation → Finite Model Theory

Keywords and phrases Model theory, finite model theory, descriptive complexity theory, logarithmic space, graph traversals

Digital Object Identifier 10.4230/OASIcs.Tannen.2024.2

Related Version *Full Version*: `https://arxiv.org/abs/2006.07067`

1 Presentation invariance

A common phenomenon in mathematics is that some property or quantity is defined in terms of some additional structure, but ends up being invariant of it. Dimension of a vector space and Euler characteristic of a manifold are important examples of this phenomenon; they are defined in terms of a given basis or simplicial complex respectively, but are invariant of the particular one chosen.

This state of affairs is very common in descriptive complexity theory. For example, suppose we want to compute the parity of a given finite set X. If we are given some linear ordering $(X, <)$, there is an inductive program computing the parity of X, but the result computed is independent of the particular ordering. Therefore, we call parity *order-invariant LFP*: computable by an LFP program with a given order, but independent of the specific choice.

The celebrated result of Immerman and Vardi [6, 11] that LFP logic captures polynomial-time queries over families of ordered finite structures can be recast as, *order-invariant* LFP logic captures polynomial-time queries over *all* finite structures. Since then, a wide array of correspondences have been identified between known complexity classes on one hand, and invariant forms of LFP, MSO, or first-order logic on the other. For example, first-order logic and LFP logic invariant in arbitrary numerical predicates captures AC^0 and $\mathrm{P}/_{\mathrm{poly}}$ respectively [9].

Our contribution. We give a novel characterization of logarithmic and nondeterministic logarithmic space queries using presentation-invariant first-order definability (Theorems 28 and 29). The presentations in question are traversals and breadth-first traversals respectively, which are certain types of linear orders on finite graphs.

[1] Corresponding author

The Provenance of Elegance in Computation – Essays Dedicated to Val Tannen.
Editors: Antoine Amarilli and Alin Deutsch; Article No. 2; pp. 2:1–2:17
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This is to our knowledge the first characterization of L or NL that does not rely on any sort of recursion or sequential computation, however limited, such as a function algebra, fixed-point logic, programming language, or automaton.

We find it fascinating and mysterious that passing from traversals to breadth-first traversals in the presentation causes a jump from L to NL in definability power. It begs the question, what other complexity classes can be characterized by certain types of graph search?

Structure of this paper. In Section 2, we discuss traversal- and breadth-first traversal-invariant definability, and show the definability of undirected and directed reachability respectively. In Section 3, we present descriptive-theoretic characterizations of L and NL (Theorems 28 and 29).

Preliminaries and notation. We assume familiarity with basic graph theory, automata theory, and model theory, including the notion of *interpretation*. We will denote graphs and other first-order structures by uppercase Roman letters. By "graph" we always mean "undirected graph;" we will say "directed graph" when we need to. We denote families of structures in a common signature by captial calligraphic letters, e.g., $\mathcal{K}$.

2 Traversals

Traversals are absolutely fundamental in computer science. They give us systematic ways of exploring a finite graph or other sort of network, and lie at the foundation of all sorts of sophisticated algorithms and techniques. Let us isolate the simplest possible version, which we call *generic graph search*, and which operates over a finite nonempty graph G.

1. Initialize a set S to some vertex in G, and repeat the following until $G \setminus S$ is empty.
2. If there is some vertex in the boundary of S, add it to S. Otherwise, add any element of $G \setminus S$ to S.

Generic graph search is nondeterministic, insofar as it does not specify which vertex to add to S. Important refinements of this algorithm include *breadth-first* and *depth-first* search, which specify additional heuristics for how to add vertices to S, without being fully deterministic.

In common parlance, the word *traversal* can refer either to the algorithm or the linear orders of G they produce, but in the current work we reserve the term "traversal," "breadth-first traversal," and "depth-first traversal" for the latter. In this paper, we do not work with depth-first traversals, but we will come back to them in the last section.

▶ **Definition 1.** *For a finite graph G, $(G,<)$ is a* traversal *(resp. breadth-first traversal, depth-first traversal) in case some instance of generic graph search (resp. breadth-first search, depth-first search) of G visits its vertices in order $<$.*

Corneil and Krueger [2] discovered that, in fact, these traversals are first-order definable in the language of ordered graphs.

▶ **Lemma 2.** *For any finite graph G and linear ordering $<$ of its vertices*

- $(G,<)$ *is a traversal iff*

$$(G,<) \models (\forall u < v < w)(uEw \to (\exists x < v)\, xEv),$$

- $(G,<)$ *is a breadth-first traversal iff*

$$(G,<) \models (\forall u < v < w)(uEw \to (\exists x < v)\, x \leq u \wedge xEv),$$

- *and* $(G,<)$ *is a depth-first traversal iff*

$$(G,<) \models (\forall u < v < w)(uEw \to (\exists x < v)\, x \geq u \wedge xEv).$$

Note that connected components of G induce intervals in a traversal. Notice also how the definitions of breadth-first traversal and depth-first traversal refine the notion of traversal in opposing ways: given a vertex v that occurs between two endpoints u and w of a single edge, v must have some prior neighbor in a plain traversal. In a breadth-first traversal, there must be some prior neighbor less than or equal to u, and in a depth-first traversal, there must be some prior neighbor greater than or equal to u.

It is an easy but important fact that

▶ **Lemma 3.** *Every finite graph admits a traversal; a fortiori, every finite graph admits both a breadth-first traversal and a depth-first traversal.*

In the present paper we characterize L and NL using traversals and breadth-first traversals respectively; it is an open question whether depth-first traversals similarly characterize some complexity class.

2.1 Traversal-invariant definability

We now present the fundamental definability-theoretic concepts in this paper. We use the standard model-theoretic notion of an *interpretation* in this definition; for details see the Appendix. If $\mathcal{K}$ is some family of structures in a common signature, by a "query over $\mathcal{K}$," we mean a boolean query, i.e., an isomorphism-closed subset of $\mathcal{K}$.[2]

▶ **Definition 4.** *Suppose $K \subseteq K^+$ are signatures, $\mathcal{K}$ is a nonempty family of K-structures, and $\mathcal{P}$ is a nonempty family of K^+-structures, such that for any $A \in \mathcal{P}$, its K-reduct $A|_K$ is in $\mathcal{K}$.*

A first-order sentence φ over $\mathcal{P}$ is $(\mathcal{K}, \mathcal{P})$-invariant *in case for any two structures A and B in $\mathcal{P}$ with the same domain, if $A|_K \cong B|_K$, then $A \models \varphi \iff B \models \varphi$.*

▶ **Definition 5.** *An n-*pointed graph *is a graph expanded with n constants. Let Γ_n be the language of n-pointed graphs, i.e., a binary relation symbol and n constant symbols.*

▶ **Definition 6.** *Let $\mathcal{G}'$ be a family of finite n-pointed graphs, $\mathcal{T}$ be the set of all expansions of structures in $\mathcal{G}'$ by any traversal, and φ be a $(\mathcal{G}', \mathcal{T})$-invariant sentence. Then for any $G \in \mathcal{G}'$, we write*

$$G \models (\mathfrak{T} <)\, \varphi$$

to indicate that for some (equivalently, any) traversal $<$ of G, $(G, <) \models \varphi$. Similarly, we write

$$G \models (\mathfrak{B} <)\, \varphi$$

if φ is $(\mathcal{G}', \mathcal{B})$-invariant where $\mathcal{B}$ is the set of all expansions by breadth-first traversals.

▶ **Definition 7.** *Let K be a signature, $\mathcal{K}$ be some family of K-structures and Q a query over $\mathcal{K}$. We say that Q is* basic traversal-invariant definable *if there exists some $n \in \mathbb{N}$, a family of finite n-pointed graphs $\mathcal{G}'$, a $(\mathcal{G}', \mathcal{T})$-invariant sentence φ, and an interpretation $\pi : \Gamma_n \to K$, such that*

[2] We will represent n-ary queries over $\mathcal{K}$ by boolean queries over the family of structures obtained by expanding every structure in $\mathcal{K}$ by any n points.

1. *π is an interpretation $\mathcal{K}$ to $\mathcal{G}'$, and*
2. *for any $A \in \mathcal{K}$,*

$$A \in Q \iff A^\pi \models (\mathfrak{T} <)\,\varphi,$$

where $\mathcal{T}$ is the set of all expansions by a traversal of all graphs in $\mathcal{G}'$.[3]

We define basic breadth-first traversal (BFT)-invariant definable *similarly. We also write $A \models ((\mathfrak{T} <)\,\varphi)^\pi$ to mean $A^\pi \models (\mathfrak{T} <)\,\varphi$.*

Note that in our definition of traversal- or BFT-invariant definability, $\mathcal{G}'$ is not required to be the family of *all* finite n-pointed graphs, though it typically will be. Note also that $\mathcal{K}$ must be a family of finite structures if there is to be a interpretation $\pi : \mathcal{K} \to \mathcal{G}'$.

▶ **Definition 8.** *Let K be a signature, $\mathcal{K}$ be some family of K-structures and Q a query over $\mathcal{K}$. Then Q is* traversal-invariant definable *(resp.* BFT-invariant definable*) if it is a boolean combination of basic traversal-invariant (resp. basic BFT-invariant) definable queries.*

We collect some important examples:

▶ **Lemma 9.** *The following queries are traversal-invariant definable over the indicated families of structures $\mathcal{K}$:*

1. *Undirected st-connectivity, over all finite 2-pointed graphs.*
2. *The family of all acyclic graphs, over all finite graphs.*
3. *The family of all bipartite graphs, over all finite graphs.*
4. *The family of even-sized finite linear orders, over all finite linear orders.*

Proof. Let $\mathcal{G}_\in$ be the family of all finite 2-pointed graphs with constants s and t. The binary reachability relation is actually definable by a single $(\mathcal{G}_\in, \mathcal{T})$-invariant sentence, which says that there is no w with no prior neighbor such that $s < w \leq t$ or $t < w \leq s$. Since components of G induce intervals in $(G, <)$, this formula asserts there is no interval separating s and t into separate connected components.

Acyclicity is similarly the spectrum of a $(\mathcal{G}, \mathcal{T})$-invariant sentence. A graph is acyclic iff, relative to any traversal, no vertex has two or more prior neighbors.

The *square* of a graph $G = (V, E)$ is the graph $G^2 = (V, E^2)$, where $E^2(x, y)$ iff x and y are connected by a path of length exactly two. Then G is bipartite iff G^2 is disconnected. Since G^2 is definable as a translation of G under an interpretation $\pi : \mathcal{G} \to \mathcal{G}$, and since connectivity is traversal-invariant definable, so is bipartiteness.

The parity of a linear order is also equivalent to the connectivity of a translation. Specifically, connect u and v by an edge iff $u = v \pm 2 \mod n$, where n is the size of the order. Then the resulting graph is either a single cycle or a union of two cycles depending on whether n is odd or even respectively. ◀

Since, e.g., connectivity and acyclicity are not Gaifman-local queries [3, 9], it follows that traversal-invariance is strictly more expressive than order-invariance.

[3] See the Appendix for the definition of notions and notations involving interpretations.

2.2 Directed reachability

Here we deal with the question of *directed* st-connectivity using BFT invariance. In fact, we will need something more than the directed graph structure, but the result will be invariant of it, an apparent asymmetry with the undirected case that will be resolved in the next section.

This construction is substantially more sophisticated than our examples above. We reduce directed reachability to an equidistance problem over undirected graphs, which we solve with the appropriate BFT-invariant sentence.

▶ **Definition 10.** *If A is a finite structure, we say that a* successor expansion (A, S) *of A is a structure of the form $(A, \min, \max, S)$, where* $\min$ *and* $\max$ *are constants and S is a* successor function *on a total order with endpoints* $\min$ *and* $\max$*.*

If K is a signature, let (K, S) be the signature of successor expansions of K-structures.[4] *If $\mathcal{K}$ is a family of finite K-structures, let $\mathcal{K}^{\mathcal{S}}$ be the family of all successor expansions of structures in $\mathcal{K}$.*

▶ **Definition 11** (Successor Invariance)**.** *For any family $\mathcal{K}$ of finite structures, let $\mathcal{K}^{\mathcal{S}}$ be the set of all successor expansions of $\mathcal{K}$. A query Q over $\mathcal{K}^{\mathcal{S}}$ is* successor-invariant *in case for any $A, B \in \mathcal{K}^{\mathcal{S}}$, if $A|_K \cong B|_K$, then $A \models Q \iff B \models Q$.*

For any $C \in \mathcal{K}$, we say $C \models (\mathfrak{S}S)\, Q$ iff some (equivalently, any) successor expansion of C satisfies Q.

▶ **Definition 12.** *Let $\mathcal{D}_n$ be the family of all finite n-pointed directed graphs, and $\mathcal{G}_n$ be the family of all finite n-pointed graphs. Let $\mathcal{D}_n^S$ be the family of all successor expansions of all finite directed n-pointed graphs.*

The interpretation ρ. We present an interpretation defined in [10] that translates directed successor graphs into undirected graphs. Let (x, y, z) be the constants of Γ_3 and (s, t) be the constants of (Γ_2, S). Consider the binary interpretation $\rho : \Gamma_3 \to (\Gamma_2, S)$ defined by

$$E^\rho(u, a; v, b) \equiv \big(S(a) = b \wedge E'(u, v)\big) \vee \big(S(b) = a \wedge E'(v, u)\big).$$

$$x^\rho = (s, \min)$$

$$y^\rho = (s, \max)$$

$$z^\rho = (t, \max),$$

where $E'(u, v)$ abbreviates $E(u, v) \vee u = v$. Then ρ is an interpretation $\mathcal{D}_2^{\mathcal{S}} \to \mathcal{G}_3$, because the predicate E^ρ is visibly symmetric. Note that ρ is also quantifier-free. We can express the st reachability problem on $D \in \mathcal{D}_2^S$ into an equidistance problem on D^ρ, cf. Figure 1 for an example. A proof of the following lemma appears in [10] and is deferred to the Appendix.

▶ **Lemma 13.** *For any graph $D \in \mathcal{D}_2^S$, there is a directed path from s to t in D iff the vertices y and z are equidistant from x in D^ρ. Even stronger, if there is no directed path from s to t in D, then either $d(x, z)$ is undefined or $|d(x, y) - d(x, z)| \geq 2$, where d indicates distance in D^ρ.*

[4] Note that there is no symbol for the order with respect to which S is a successor function in the signature (K, S).

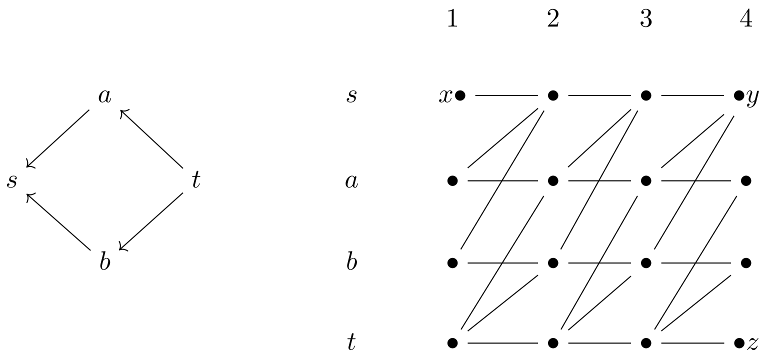

Figure 1 A directed graph to the left, with its undirected ρ-translation to the right. There is an edge between (u, i) and $(v, i+1)$ on the right exactly when $u = v$ or there is a directed edge $u \to v$ on the left. Consequently, there is a directed path $s \rightsquigarrow t$ on the left exactly when y and z are equidistant from x on the right. Here there is no such path $s \rightsquigarrow t$, $d(x, y) = 3$, and $d(x, z) = 7$.

▶ **Definition 14.** *Let $\mathcal{G}_3'$ be the family of finite 3-pointed undirected graphs with constants x, y, z such that x and y lie in the same connected component and, if z does as well, then $|d(x, y) - d(x, z)| \neq 1$.*

By Lemma 13, ρ is in fact an interpretation $\mathcal{D}_2^S \to \mathcal{G}_3'$.

Breadth-first traversals and quasi-levels. On a graph with a distinguished source for each connected component, vertices are naturally partitioned into *levels* according to their distance from their respective source. If we fix a BFT of a graph, and let the source of each connected component be its least element, then the resulting levels induce intervals in that traversal.[5] Moreover, every edge of the graph is either within levels or between adjacent levels. The least neighbor of every vertex (except the source) is in the previous level.

It is probably impossible to recognize when two nodes are in the same level using first-order logic on graphs, even given a BFT. However we can do almost as well.

▶ **Definition 15.** *Let $(V, E, <)$ be a finite graph expanded by a breadth-first traversal. A* quasi-level *is a nonempty interval I of (V, E) such that $w \in I \iff p(w) < v \leq w$, where v is the least element of I and $p(w)$ the least neighbor of w (cf. Figure 2).*

Observe that it is easy to define when two vertices v and w occur in a common quasi-level, by the formula

$$(p(w) < v \leq w) \vee (p(v) < w \leq v).$$

Notice that if two vertices occur in a common quasi-level, then their distances from their (necessarily common) source cannot differ by more than 1. If two vertices occur in no common quasi-level, then either they are in different connected components, or the distances from their common source cannot be equal.

[5] Recall that connected components induce intervals of a traversal, so it suffices to observe that levels induce intervals within connected components.

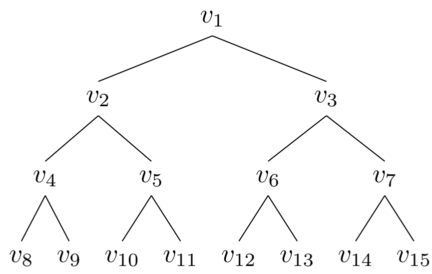

Figure 2 $\{v_2, v_3\}$, $\{v_3, v_4, v_5\}$, $\{v_6, \ldots, v_{11}\}$, and $\{v_{13}, v_{14}, v_{15}\}$ are quasi-levels but $\{v_5, \ldots, v_{11}\}$ and $\{v_4, v_5, v_6\}$ are not. In the first counterexample, $v_{11} \in I$, but $p(v_{11}) = v_5$, the least element of I, and in the second counterexample, $p(v_7) < v_4 < v_7$ and v_4 is the least element of I, but $v_7 \notin I$.

The interpretation τ. We define a 2-dimensional interpretation $\tau : \Gamma_6 \to \Gamma_3$. Let

$$(x_1, y_1, z_1, x_2, y_2, z_2)$$

be the constants in Γ_6 and (x, y, z) be the constants in Γ_3. Given $G \in \mathcal{G}_3$, the domain of G^τ consists of "two copies" of G, which we achieve by $\partial^\tau(u, v) \iff v = x \lor v = y$. Within each copy, we inherit the edge relation from G, and we let, e.g., x_i be the vertex corresponding to x in copy i. We do not put any edges between the two copies except for connecting x_1 and x_2. Notice that τ is quantifier-free.

▶ **Definition 16.** *Let $\mathcal{G}'_6$ be $\{G^\tau : G \in \mathcal{G}'_3\}$.*

Then (by definition), τ is an interpretation $\mathcal{G}'_6 \to \mathcal{G}'_3$. Moreover,

▶ **Theorem 17.** *Let $\mathcal{B}$ be the set of all expansions of graphs in $\mathcal{G}'_6$ by a breadth-first traversal. There is a $(\mathcal{G}'_6, \mathcal{B})$-invariant formula ψ such that for any $(G, x, y, z) \in \mathcal{G}'_3$,*

$$d(x, y) = d(x, z) \implies G^\tau \models (\mathfrak{B} <)\, \psi$$

$$|d(x, y) - d(x, z)| \geq 2 \implies G^\tau \models \neg(\mathfrak{B} <)\, \psi,$$

where the second case also contains all those graphs where x, y, and z are not all connected.

(The proof is deferred to the Appendix.)

By composing the interpretation ρ with the interpretation τ, we see that for any successor expansion of a finite 2-pointed directed graph $D \in \mathcal{D}^S_2$, there is a path from s to t in D if and only if $D^{\rho\tau} \models (\mathfrak{B} <)\, \psi$. Hence,

▶ **Corollary 18.** *The directed reachability query is BFT-invariant definable over $\mathcal{D}^S_2$.*

3 Descriptive Complexity

In this section we obtain the main results of this paper: a characterization of deterministic and nondeterministic logarithmic space by traversal and breadth-first traversal invariance quantifiers respectively.

3.1 Multihead finite automata

A *nondeterministic multihead finite automaton* (NMFA) is an automaton with a single tape, finitely many heads on that tape, and a finite control. Unlike a Turing machine, the tape is not infinite; rather, it is initialized to the input string plus two special characters on either side to mark the left and right endpoints. Also unlike a Turing machine, the heads cannot write, they can only move left, right, or stay put depending on which characters they are reading. A single state is designated as accepting; if the computation enters this state then we say it halts. The language of an NMFA is exactly the set of strings it halts on.

Formally, an NMFA consists of a set Q of states, some number $k \in \mathbb{N}$ of heads, an input alphabet Σ, a start state $q_0 \in Q$, an accept state $q_f \in Q$, and a transition relation δ which relates k-tuples in $\Sigma \cup \{\triangleright, \triangleleft\}$ with $\{-1, 0, 1\}^k$. If any head is reading the left (respectively right) endpoint character, no subsequent transition may move that head right (respectively left). Futhermore, if the current state is q_f, then the transition relation moves all heads to the left (if possible) or fixes them if they are already at the left endpoint.

A *configuration* of an NMFA consists of the input string, the current state, and the location of the heads. The transition relation induces a relation on the space of configurations in the natural way. The *intial configuration* is the one in which the state is q_0 and all heads are at the left. The *final configuration* is the same but with state q_f. By the stipulation of the transition relation, if an NMFA enters q_f, then it will always enter the final configuration.

The *configuration graph* of an NMFA on a particular input x is a 2-pointed directed graph whose vertices are the set of configurations on x and whose edge relation is the graph of the relation induced by the transition function. The source and sink are the initial and final configurations respectively.

Strings and pointed graphs as structures. Let Γ_2 be the language of 2-pointed graphs, and let (Γ_2, S) be the language of 2-pointed successor graphs, with two (additonal) constants min and max, and a successor function.

Let Σ be a finite alphabet. We think of a string $x = x_0 x_1 \dots x_{n-1}$ in $\Sigma^\star$ as a finite structure with domain $\{0, 1, \dots, n-1\}$, a predicate σ for each $\sigma \in \Sigma$ with semantics

$$(\forall i < n)\ x \models \sigma(i) \iff x_i = \sigma,$$

constants min and max naming 0 and $n-1$, and finally a successor function taking index i to index $i+1$.[6]

We henceforth overload the meaning of Σ to indicate not only an alphabet, but also the signature of strings in that alphabet, so that the terms "finite Σ-structure" and "member of $\Sigma^\star$" denote the same objects.

Crucial to our work is that for a fixed NMFA, there is an interpretation that takes an input string and translates it into the associated configuration graph.

► **Theorem 19.** *For every NMFA $\mathcal{M}$ with alphabet Σ there is an interpretation $\pi : \Gamma_2 \to \Sigma$ such that for every sufficiently long string $x \in \Sigma^\star$, x^π is isomorphic to the configuration graph of $\mathcal{M}$ on input x.*

Furthermore, we can expand π to an interpretation $\pi : (\Gamma_2, S) \to \Sigma$, so that x^π is a successor expansion of the above configuration graph.

Moreover, π can be made quantifier-free.

(The proof is deferred to the Appendix)

► **Definition 20.** *An NMFA $\mathcal{M}$ is* symmetric *(SMFA) in case, for any input x, the configuration graph of $\mathcal{M}$ on x is undirected.*

[6] This is a common, "folklore," method of representing strings as structures. Often one takes a total ordering $<$ over the indices of a string instead of the successor function (see Libkin [7]), but for our purposes, either will work.

Computability by NMFAs is known to capture exactly nondeterministic logarithmic space (NL) [4], and computability by SMFAs captures at least logarithmic space (L).[7]

3.2 Capturing L and NL

Canonical encodings. For any finite structure A, any linear order $(A, <)$, and any fixed alphabet Σ of size at least 2, there is a *canonical encoding* of $(A, <)$ as a string in $\Sigma^\star$.

This construction is the foundation of all results in descriptive complexity, and can be found in numerous texts, e.g., [7]. We will not repeat it here. We do note, however, that any successor expansion (A, S) of A induces a linear order – hence every successor expansion of any finite structure has a canonical encoding.

Even more importantly, this canonical encoding is *definable* as the translation induced by a quantifier-free interpretation. The details are complicated, but can be found in Section 9.2 of [7].

▶ **Theorem 21.** *For every signature L, there is a quantifier-free interpretation $\mu : \Sigma \to (L, S)$ such that for every successor expansion (A, S) of any finite L-structure A, $(A, S)^\mu$ is the canonical encoding of (A, S).*

We now state the definition of a complexity-bounded query over finite structures, for which we need to imagine models of computation that take finite structures as input. We follow the standard method in descriptive complexity, which is to take a model of computation that operates on strings, and feed it the encoding $(A, S)^\mu$ of a structure A. Of course this encoding is not canonical given only A; for a well-defined query, we demand that the result of the computation is invariant of the particular expansion (A, S).

Now we are in a position to state:

▶ **Theorem 22.** *For every signature L and every logarithmic space query Q over finite L-structures, there is a quantifier-free interpretation $\gamma : \Gamma_2 \to (L, S)$ such that for every sufficiently large finite L-structure A,*

$$A \in Q \iff A \models (\mathfrak{S} S)\,((\mathfrak{T} <)\,\varphi)^\gamma,$$

where $(\mathfrak{T} <)\,\varphi$ is the sentence in the language of 2-pointed ordered graphs asserting that the distinguished vertices are connected.

Proof. Let $\mu : \Sigma \to (L, S)$ be the interpretation given by L in Theorem 21, let $\mathcal{M}$ be an SMFA deciding Q, let $\pi : \Gamma_2 \to \Sigma$ be the associated interpretation from Theorem 19, and let $\gamma = \mu\pi$. Fix a finite L-structure A and an arbitrary successor expansion (A, S).

Then $\mathcal{M}$ accepts the string $(A, S)^\mu$ iff $A \in Q$. But, $(A, S)^{\mu\pi} = (A, S)^\gamma$ is the configuration graph of $\mathcal{M}$ on $(A, S)^\mu$, so $\mathcal{M}$ accepts $(A, S)^\mu$ just in case the distinguished vertices of $(A, S)^\gamma$ are connected. In other words,

$$A \in Q \iff (A, S)^\gamma \models (\mathfrak{T} <)\,\varphi.$$

Therefore,

$$A \in Q \iff (A, S) \models ((\mathfrak{T} <)\,\varphi)^\gamma,$$

and since the right hand side is independent of the particular successor expansion,

$$A \in Q \iff A \models (\mathfrak{S} S)\,((\mathfrak{T} <)\,\varphi)^\gamma.$$ ◀

[7] In [1], Axelsen considers the more restrictive *reversible* MFAs, which are both deterministic and backwards deterministic, and shows that they capture L. It is plausible that this might allow us to strengthen our results even further.

▶ **Theorem 23.** *For every signature L and every* NL *query Q over finite L-structures, there is a quantifier-free interpretation $\gamma : \Gamma_6 \to (L, S)$ such that for every sufficiently large finite L-structure A,*

$$A \in Q \iff A \models (\mathfrak{S}S)\,((\mathfrak{B} <)\,\psi)^{\gamma},$$

where ψ is the sentence from Theorem 17 in the language $(\Gamma_6, <)$.

Proof. Let $\mu : \Sigma \to (L, S)$ be the interpretation given by L in Theorem 21, let $\mathcal{M}$ be an NMFA deciding Q, let $\pi : (\Gamma_2, S) \to \Sigma$ be the associated interpretation from Theorem 19. Recall the interpretations $\rho : \Gamma_3 \to (\Gamma_2, S)$ and $\tau : \Gamma_6 \to \Gamma_3$ from Section 2.2. Finally, let $\gamma = \mu\pi\rho\tau$. Fix a finite L-structure A and an arbitrary successor expansion (A, S).

Then $\mathcal{M}$ accepts the string $(A, S)^{\mu}$ iff $A \in Q$. But $\mathcal{M}$ accepts $(A, S)^{\mu}$ just in case there is a path from source to sink over the graph $(A, S)^{\mu\pi}$. By Corollary 18, this occurs just in case $(A, S)^{\mu\pi\rho\tau} \models (\mathfrak{B} <)\,\psi$. But $(A, S)^{\mu\pi\rho\tau} = (A, S)^{\gamma}$.

Since the above is independent of the particular successor expansion S,

$$A \in Q \iff A \models (\mathfrak{S}S)((\mathfrak{B} <)\,\psi)^{\gamma},$$

which completes the proof. ◀

3.3 Logspace-computable traversals

In the other direction, we want to show that traversals and breadth-first traversals are computable in L and NL respectively. These constructions rely on the computability in logarithmic space of undirected *st*-connectivity, and furthermore on the existence of logarithmic space *universal exploration sequences* [8].

Given an ordered graph G and a vertex v, it is possible to construct, in logarithmic space, the index of the least vertex u in the connected component of v. Simply iterate through the vertices of G in order, testing connectivity with v, until we find a vertex that is connected.

▶ **Theorem 24.** *There is a logarithmic space Turing machine which, for every finite ordered graph $(G, <)$, computes a traversal $(G, \prec)$ in the following sense: given the canonical encoding of $(G, <)$ and (indices of) two of its vertices v and w, accepts or rejects according to whether $v \prec w$.*

Proof. Given two vertices v and w in G, first test whether they are in the same connected component. If not, let v_0 and w_0 be the least elements in the connected components of v and w respectively, and compare v and w according to whether $v_0 < w_0$.

Otherwise, let v_0 be the least element of their common connected component. If $n = |G|$, construct (using space logarithmic in n) a universal exploration sequence, and explore the connected component of v and w according to that sequence starting with v_0. Let $v \prec w$ iff the first occurrence of v precedes the first occurrence of w.

We must show $(G, \prec)$ is a traversal. Notice that connected components induce intervals. If v is not the least vertex in some connected component, then its first occurrence in the universal exploration sequence has some immediate predecessor u which is a neighbor. Therefore, in the traversal, $u \prec v$; hence, v has some preceding neighbor. ◀

Canonical BFT of an ordered graph. Unlike the case of ordinary traversals, where the traversal $(G, \prec)$ of $(G, <)$ depends on some family of universal exploration sequences, we will define a canonical breadth-first traversal $(G, \prec_B)$ of an ordered graph $(G, <)$ and show that it can be computed in nondeterminstic logspace.

▶ **Definition 25.** *Given a finite ordered graph $(G,<)$ and vertices $v,w \in G$, let v_0 and w_0 be the $<$-least elements of the connected components of v and w respectively. Let $<^\star$ be the ordering on finite sequences of vertices that orders them* first *by length, and* then *lexicographically. Let $\vec{v}$ be the $<^\star$-least path from v_0 to v. Then,*
1. *if $v_0 \neq w_0$, then $v \prec_B w \iff v_0 < w_0$,*
2. *if $v_0 = w_0$ then $v \prec_B w \iff \vec{v} <^\star \vec{w}$.*

▶ **Lemma 26.** *For any finite ordered graph $(G,<)$, $(G,\prec_B)$ is a breadth-first traversal.*

(Proof deferred to appendix.)

▶ **Theorem 27.** *There is a logarithmic space nondeterministic Turing machine which, on input a finite ordered graph $(G,<)$ and vertices $v,w \in G$, decides whether or not $v \prec_B w$.*

Proof. As in the proof of Theorem 24, given two vertices v and w, first test whether $v_0 = w_0$ (i.e., whether they're in the same connected component). If not, decide $v \prec_B w$ according to whether $v_0 < w_0$.

Otherwise we argue that we can construct the sequence $\vec{v} = (v_0,\ldots,v_{\ell-1},v)$ in the following sense: given an index for v_i, we can test whether it's equal to v; if not, we can construct the index of v_{i+1}, all in logarithmic space.

If we can do this, then we decide $v \prec_B w$ by comparing $\vec{v} <^\star \vec{w}$. First we compare their lengths: we simultaneously construct (v_{i+1},w_{i+1}) from (v_i,w_i), until the first index is v or the second is w. Unless this happens at the same stage, we are done. (Since (v_{i+1},w_{i+1}) overwrites (v_i,w_i), this remains in logarithmic space.)

Otherwise, we start over, and simultaneously construct (v_i,w_i) until we (necessarily) find the first index at which they differ. Then we decide $v \prec_B w$ according to which is larger.

It remains to show how to construct v_{i+1} from v_i. Orient all edges in G so that they increase distance from v_0. Then v_{i+1} is the $<$-least vertex x such that there is an edge (v_i,x) and a directed path (x,v). Since we can compute directed reachability in nondeterminstic logarithmic space, we can find v_{i+1} in nondeterminstic logarithmic space as well. ◀

At this point we are ready to state two of the central results of this paper.

▶ **Theorem 28.** *The following are equivalent:*
1. *Q is a logspace-decidable query over finite K-structures.*
2. *There is a quantifier-free interpretation $\pi : \Gamma_2 \to (K,S)$ such that for all sufficiently large finite K-structures A,*
$$A \in Q \iff A \models (\mathfrak{S}S)\,((\mathfrak{T}<)\,\varphi)^\pi,$$
where φ is the formula expressing undirected st-connectivity.
3. *There is a traversal-invariant definable query R over finite (K,S) structures such that for any finite K-structure A,*
$$A \in Q \iff A \models (\mathfrak{S}S)\,R$$

Proof. Implication $1 \Rightarrow 2$ is exactly Theorem 22. Implication $2 \Rightarrow 3$ is immediate, as $((\mathfrak{T}<)\,\varphi)^\pi$ is by definition a traversal-invariant definable query, and traversal-invariant queries are closed under finite differences. It remains to show $3 \Rightarrow 1$.

It suffices to show that the traversal-invariant definable query R is logspace computable over finite (K,S) structures, as given an encoding of a K-structure A, a logspace Turing machine can always compute a successor relation on the domain of A, by using the particular encoding in which A is presented.

Since logspace-computable queries are closed under boolean combinations, it suffices to show that any basic traversal-invariant definable query is logspace computable. Since logspace computable queries are closed under elementary interpretations, it suffices to show that for any class $\mathcal{G}'$ of finite graphs, every $(\mathcal{G}', \mathcal{T}')$-invariant formula is logspace computable over graphs in $\mathcal{G}'$, where $\mathcal{T}'$ is the family of all expansions of graphs in $\mathcal{G}'$ by traversals.

But for this, it suffices to show that any first-order sentence in the language of ordered graphs is logspace computable given an encoding of a finite graph, where the order is the traversal defined in Theorem 24. Since logspace queries are closed under first-order combinations, it suffices to check that given any encoding of a graph and two vertices therein, we can test whether they are equal, test whether they are connected by an edge, or compare them according to the canonical traversal.

The first two are true, and the last is exactly Theorem 24. ◀

By replacing "L" by "NL" and "traversal" by "breadth-first traversal" throughout, we get

▶ **Theorem 29.** *The following are equivalent:*

1. *Q is an nlogspace-decidable query over finite K-structures.*
2. *There is a quantifier-free interpretation $\pi : \Gamma_6 \to (K, S)$ such that for all sufficiently large finite K-structures A,*

$$A \in Q \iff A \models (\mathfrak{S}S)\,((\mathfrak{B} <)\,\psi)^{\gamma},$$

where ψ is the sentence of Theorem 17.

3. *There is a breadth-first traversal-invariant definable query R over finite (K, S) structures such that for any finite K-structure A,*

$$A \in Q \iff A \models (\mathfrak{S}S)\,R$$

Structures with successor. Suppose the signature K contains the unary function symbol S, and that $\mathcal{K}$ is a family of finite K-structures in which S is interpreted by a successor function. Then the quantifier $(\mathfrak{S}S)$ in any successor-invariant query $(\mathfrak{S}S)\,R$ is superfluous over $\mathcal{K}$; i.e., for any $A \in \mathcal{K}$,

$$A \models R \iff A \models (\mathfrak{S}S)\,R.$$

The reason is that the interpretation of S in R is independent of the particular successor function on A that we choose, so we might as well choose the one native to A.

In particular, for such families $\mathcal{K}$, we can drop the $(\mathfrak{S}S)$ quantifier from the traversal- or breadth-first traversal-invariant queries R of Theorems 28 and 29. In particular, let us take the case of strings over a finite alphabet Σ, which are the original setting for logspace and nlogspace queries, and also successor structures as described in Section 3. Then we have

▶ **Corollary 30.** *For any family $Q \subseteq \Sigma^\star$,*

1. *Q is logspace-decidable iff there is a traversal-invariant definable query R such that for every string $x \in \Sigma^\star$, $x \in Q \iff x \models R$, and*
2. *Q is nlogspace-decidable iff there is a breadth-first traversal-invariant definable query R such that for every string $x \in \Sigma^\star$, $x \in Q \iff x \models R$.*

3.4 Discussion and open questions

Our results are the first presentation-invariant characterizations of L and NL, and, to our knowledge, the largest known complexity classes characterized by first-order logic extended by invariant definability of an elementary class of presentations. They demonstrate the surprising power of interpretations (even quantifier-free ones!) and establish a new foundational correspondence between graph traversals and complexity classes.

The elephant in the room is whether *depth-first traversal invariance* captures a meaningful complexity class, like polynomial time. While we have been able to find depth-first invariant definitions of certain suggestive queries (like *vertex-avoiding paths*), we still do not have very strong evidence one way or the other. More generally, there are a variety of graph traversals and a variety of associated presentations (such as the ancestral relation of the traversal tree) which might correspond to interesting complexity classes.

Finally, we have extended these notions of definability to arbitrary infinite structures by requiring that the underlying order be well-founded. (Since well-orders are not elementarily definable, this circumvents the usual "Beth definability" obstacle to studying presentation invariance over infinite structures.) Whereas separating traversal-invariant from BFT-invariant definability over classes of finite structures requires separating L and NL, it is plausibly easier to separate them over arbitrary classes, and it is plausible that this will inform the finite case. This work is ongoing.

References

1 Holger Bock Axelsen. Reversible multi-head finite automata characterize reversible logarithmic space. In Adrian-Horia Dediu and Carlos Martín-Vide, editors, *Language and Automata Theory and Applications*, pages 95–105, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

2 Derek Corneil and Richard Krueger. A unified view of graph searching. *SIAM J. Discret. Math.*, 22(4):1259–1276, July 2008.

3 Yuri Gurevich. Toward logic tailored for computational complexity. In M. M. Ricther et al., editor, *Computation and Proof Theory, Lecture Notes in Mathematics 1104*, pages 175–216. Springer-Verlag, 1984.

4 Juris Hartmanis. On non-determinancy in simple computing devices. *Acta Informatica*, 1(4):336–344, December 1972.

5 Wilfrid Hodges. *Model Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1993.

6 Neil Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

7 Leonid Libkin. *Elements of Finite Model Theory*. Springer, Berlin, Heidelberg, 2004.

8 Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4):17:1–17:24, September 2008.

9 Nicole Schweikardt. A short tutorial on order-invariant first-order logic. In *CSR 2013*, June 2013.

10 Martin Tompa. *Introduction to Computational Complexity*. Lecture notes, 1991. URL: `https://homes.cs.washington.edu/~tompa/papers/532.pdf`.

11 Moshe Vardi. The complexity of relational query languages. In *STOC82*, pages 137–146, 1982.

A Interpretations and change of signature

We review the basic definitions behind interpretations, following the exposition of Hodges [5], except that we also allow for functional signatures (see below). There is no new mathematical content here; however, getting the definitions and terminology straight is terribly important, since we use interpretations extensively.

▶ **Definition 31.** *Let L and K be signatures and $k \in \mathbb{N}$. An* elementary k-ary interpretation *$\pi : L \to K$ is a first-order K-formula $\partial^\pi(\bar{x})$, for each constant symbol $c \in L$ a variable-free K-term c^π, and for each relation symbol $r \in L$, a first-order K-formula $r^\pi(\bar{x}_1, \ldots, \bar{x}_n)$, where n is the arity of r, and the length of each tuple throughout is k.*

An interpretation is quantifier-free *in case ∂^π and each r^π is quantifier-free.*

▶ **Definition 32.** *Given an elementary k-ary interpretation $\pi : L \to K$ and a first-order L-term or L-formula α, its* translation *α^π, is a K-formula given by the following recursion:*

1. *If α is a variable x, then α^π is a k-tuple of (distinct) variables $\bar{x}$.*
2. *If α is a constant symbol c, then α^π is c^π.*
3. *If α is the atomic formula $r(\alpha_1, \ldots, \alpha_n)$, then α^π is $r^\pi(\alpha^\pi, \ldots, \alpha^\pi)$,*
4. *If α is a boolean combination of formulas ϑ, then α^π is the same boolean combination of formulas ϑ^π,*
5. *If α is $\exists x\, \vartheta$, then α^π is the formula $\exists \bar{x}\, \partial^\pi(\bar{x}) \wedge \vartheta^\pi$, and*
6. *If α is $\forall x\, \vartheta$, then α^π is the formula $\forall \bar{x}\, \partial^\pi(\bar{x}) \to \vartheta^\pi$.*

In the definition below, $\partial^\pi[A^k]$ denotes the subset of A^k on which ∂^π holds.

▶ **Definition 33.** *Suppose $\pi : L \to K$ is an interpretation and A is a K-structure. Then the* π-translation *A^π is the L-structure with domain $\partial^\pi[A^k]$ with the denotation of λ given by λ^π, for each $\lambda \in L$.*

(Note that even though the arity of λ^π is nk as a K-formula, it defines an n-ary relation over A^π, whose elements are k-tuples of A.)

Functional signatures. In a very particular case (see *successor expansions*, Definition 10) we will want to consider signatures with function symbols, and exactly once (Theorem 19), we will want to define an interpretation $\pi : L \to K$ where L has some function symbol $f(x_1, \ldots, x_n)$. In this case $f^\pi(\vec{x}_1, \ldots, \vec{x}_n)$ is a *definition by cases*, where each case is a first-order K-formula, and the *definiens* inside each case is a k-tuple of K-terms in the free variables $(\vec{x}_1, \ldots, \vec{x}_n)$, where k is the arity of π. In a quantifier-free interpretation, each case must be a quantifier-free K-formula.

Functional signatures also generalize signatures with constants, which are nullary function symbols. For a constant symbol c, c^π is a definition by cases, where each case is a k-tuple of variable-free K-terms.

It is common practice in finite model theory to replace functions by their graph relations, thus working with purely relational signatures. The only reason for considering functional signatures here is to make certain interpretations quantifier free (cf. Theorems 28 and 29); in the purely relational setting, these interpretations would contain quantifiers.

Injective interpretations. Usually an interpretation will also contain a first-order K-formula $eq^\pi(\bar{x}, \bar{y})$ defining when we regard two k-tuples as equal. (For example, when interpreting rational numbers by pairs of integers, we say $(a, b) = (c, d) \iff ac - bd = 0$.) In case eq^π is simply equality of tuples (as above), π is called an *injective interpretation.* Here we do not deal with any interpretations with a nontrivial equivalence relation. Therefore, it is convenient to drop the word "injective" and simply refer to interpretations.

▶ **Lemma 34** (Fundamental property of interpretations)**.** *Suppose that $\pi : L \to K$ is an elementary interpretation. Then for every K-structure A, every n-ary L-sentence φ, and every $\bar{x}_1, \ldots, \bar{x}_n$ in the domain $\partial^\pi[A^k]$ of A^π,*

$$A \models \varphi^\pi(\bar{x}_1, \ldots, \bar{x}_n) \iff A^\pi \models \varphi(\bar{x}_1, \ldots, \bar{x}_n).$$

Note that on the left-hand side, $(\bar{x}_1, \ldots, \bar{x}_n)$ is regarded as an nk-tuple of elements in A, and on the right-hand side, it is regarded as an n-tuple of elements in A^π.

▶ **Definition 35.** *Suppose that $\pi : L \to K$, $\mathcal{L}$ is a class of L-structures, and $\mathcal{K}$ is a class of K-structures. Then π is an* interpretation $\mathcal{K} \to \mathcal{L}$ *in case for every $A \in \mathcal{K}$, $A^\pi \in \mathcal{L}$.*

Properties of interpretations ρ and τ

Lemma 13. For any graph $D \in \mathcal{D}_2^S$, there is a directed path from s to t in D iff the vertices y and z are equidistant from x in D^ρ. Even stronger, if there is no directed path from s to t in D, then either $d(x,z)$ is undefined or $|d(x,y) - d(x,z)| \geq 2$, where d indicates distance in D^ρ.

Proof of Lemma 13. (Adapted from [10]) Fix a graph D and let n be the number of vertices in D. Identify the vertices of D with $\{0, 1, \ldots, n-1\}$ such that $S(i, i+1)$. Then in D^ρ, there is a path

$$x = (s,0) - (s,1) - \cdots - (s, n-1) = y,$$

of length $n-1$, and this is moreover the distance between x and y, by considering the second coordinate.

If t is reachable from s in D, then that must be witnessed by some directed path $(s = r_0 \to r_1 \to \cdots \to r_{\ell-1} = t)$ of length $\ell \leq n$. Then

$$(r_0, 0) - (r_1, 1) - \cdots - (r_{\ell-1}, \ell-1) - (r_{\ell-1}, \ell) - \cdots - (r_{\ell-1}, n-1)$$

is a path in D^ρ from x to z of length exactly $n-1$. Again by considering the second coordinate, we can see that there is no shorter path. Hence y and z are equidistant from x.

Conversely, suppose that there were a path in D^ρ from x to z in D^ρ of length exactly $n-1$. Then it must be of the form

$$(u_0, 0) - (u_1, 1) - \cdots - (u_{n-1}, n-1),$$

where $u_0 = s$, $u_{n-1} = t$, and for each i, either $u_i = u_{i+1}$ or $u_i \to u_{i+1}$ in D. Hence the u_i witness a directed path from s to t.

Moreover, observe the parity of the second coordinate in any path must alternate. Hence, the length of any path from $(s,0)$ to $(t, n-1)$ must be equal to $n-1$ modulo 2. Therefore, if there is no directed path from s to t in D, then in D^ρ then any path from x to z in D^ρ must have length at least $n+1$. This is at least 2 greater than $d(x,y)$, which is $n-1$. ◀

Theorem 17. Let $\mathcal{B}$ be the set of all expansions of graphs in $\mathcal{G}_6'$ by a breadth-first traversal. There is a $(\mathcal{G}_6', \mathcal{B})$-invariant formula ψ such that for any $(G, x, y, z) \in \mathcal{G}_3'$,

$$d(x,y) = d(x,z) \implies G^\tau \models (\mathfrak{B} <)\, \psi$$

$$|d(x,y) - d(x,z)| \geq 2 \implies G^\tau \models \neg(\mathfrak{B} <)\, \psi,$$

where the second case also contains all those graphs where x, y, and z are not all connected.

Proof of Theorem 17. Let ψ assert that all six constants $(x_1, \ldots, z_2)$ occur in the same connected component; moreover, if $x_1 < x_2$, then y_2 and z_2 occur in the same quasi-level, and if $x_2 < x_1$, then y_1 and z_1 occur in the same quasi-level. (To show that ψ is invariant, it suffices to show that ψ is correct.)

Fix a graph $(G, x, y, z) \in \mathcal{G}'_3$ such that $d(x, y) = d(x, z)$. Consider its translation, and expand this by an arbitrary breadth-first traversal. We may assume that all constants $(x_1, \ldots, z_2)$ lie in the same connected component; otherwise ψ evaluates to false, which is correct as not all of (x, y, z) are connected.

Suppose that $x_1 < x_2$. Let w be the least element of $<$ in the connected component of x_1. Then w must be in G_1, so any path from w to y_2 or z_2 must pass through the edge (x_1, x_2). Hence,

$$|d(x, y) - d(x, z)| = |d(x_2, y_2) - d(x_2, z_2)| = |d(w, y_2) - d(w, z_2)|.$$

In other words, the desired quantity is exactly the difference in distance between y_2 and x_2 to the source. We know that this difference is either equal to 0 or at least 2, and ψ correctly distinguishes these two cases by testing whether y_2 and z_2 occur in the same quasi-level.

Similarly, if $x_2 < x_1$, ψ distinguishes $|d(x, y) - d(x, z)| = 0$ from $|d(x, y) - d(x, z)| \geq 2$ by testing whether y_1 and z_1 occur in the same quasi-level. ◀

B Defining configuration graphs by interpretation

Theorem 19. For every NMFA $\mathcal{M}$ with alphabet Σ there is an interpretation $\pi : \Gamma_2 \to \Sigma$ such that for every sufficiently long string $x \in \Sigma^\star$, x^π is isomorphic to the configuration graph of $\mathcal{M}$ on input x. Furthermore, we can expand π to an interpretation $\pi : (\Gamma_2, S) \to \Sigma$, so that x^π is a successor expansion of the above configuration graph. Moreover, π can be made quantifier-free.

Proof of Theorem 19. Let k be the number of heads in $\mathcal{M}$. Then $k+1$ will be the dimension of π. The domain of the interpretation ∂_π just stipulates that the first coordinate is less than q, the number of states of $\mathcal{M}$.

We can now establish a bijection between the domain of π and configurations of $\mathcal{M}$ with input x, for any string x such that $|x| \geq q$. A configuration is simply specified by current state and the location of the heads, which correspond to the first and remaining k coordinates of the domain respectively. Since x is sufficiently long, there are enough choices in the first coordinate for all states of $\mathcal{M}$.

Let $\vec{u}$ and $\vec{v}$ be arbitrary configurations of $\mathcal{M}$ on input an arbitrary string of length at least q. We want to define $E^\pi(\vec{u}, \vec{v})$ to hold just in case the configuration $\vec{v}$ is reachable from $\vec{u}$ in one step. This is definable by a boolean combination of formulas of the following form:

1. u_i is the minimum or maximum index,
2. $\sigma \in \Sigma$ is the character at index, and
3. indices u_i and v_i are identical or adjacent.

Each of these formulas is quantifier-free definable in the language Σ, by e.g.,

1. $u_i = 0$ or $u_i = n - 1$,
2. $\sigma(u_i)$, and
3. $u_i = v_i$ or $S(u_i) = v_i$ or $u_i = S(v_i)$

respectively, where 0 and $n - 1$ are aliases for min and max respectively. Finally,

$$s^\pi = (0, 0, \ldots, 0), \ t^\pi = (S(0), 0, \ldots, 0).$$

This is because all heads are at the left in the initial or final configuration, 0 is the start state, and 1 is the halt state.

Now for any string x of length at least q, not only is the domain of x^π in bijection with the configurations of $\mathcal{M}$ on x, but relative to this bijection π_E defines the graph of "next," and π_s and π_t are the initial and final configurations respectively. Hence x^π as a structure is isomorphic to the configuration graph of $\mathcal{M}$ on input x.

To expand π to an interpretation from Γ_2^S, we need to define a successor function on $(k+1)$-tuples of indices, given a successor function on indices. This is easy to do by mimicking the standard "increment-by-one" algorithm on numbers written in some fixed radix.[8] ◀

First-order definitions of traversals

Lemma 26. For any finite ordered graph $(G, <)$, $(G, \prec_B)$ is a breadth-first traversal.

Proof of Lemma 26. Connected components of G induce intervals of $(G, \prec_B)$, so it suffices to assume that G is connected. Let v_0 be the least element of G (unambiguously with respect to either order).

Suppose v is a non-minimal vertex and let $\vec{v} = (v_0, v_1, \ldots, v_{\ell-1}, v)$. Since $<^\star$-least shortest paths are closed under prefixes, $v_i \prec_B v$ for each v; in particular, $|\vec{v}_{\ell-1}| = \ell - 1$.

Let u be the $\prec_B$-least neighbor of v, and $\vec{u}$. Since $u \preceq_B v_{\ell-1}$, $|\vec{u}| \leq \ell - 1$. Since $(\vec{u}, v)$ is a path from v_0 to v of length at most ℓ, and since ℓ is the distance from v_0 to v, $|\vec{u}| = \ell - 1$.

Since u and $v_{\ell-1}$ are the same distance from v_0, we have

$$\vec{u} \leq^\star (v_0, \ldots, v_{\ell-1}) \wedge (v_0, \ldots, v_{\ell-1}, v) \leq^\star (\vec{u}, v).$$

Therefore $\vec{u} = (v_0, \ldots, v_{\ell-1})$. In particular, $u = v_{\ell-1}$.

Finally, suppose that v and w are arbitrary non-minimal vertices of G, and that $v \prec_B w$. Let $v_\dagger$ and $w_\dagger$ be the second-to-last elements of $\vec{v}$ and $\vec{w}$ respectively. Then $v_\dagger$ and $w_\dagger$ are the $\prec_B$-least neighbors of v and w, so it suffices to show that $v_\dagger \preceq_B w_\dagger$.

However, $\vec{v} = (\vec{v}_\dagger, v)$ and $\vec{w} = (\vec{w}_\dagger, w)$. Since $\vec{v} <^\star \vec{w}$, $\vec{v}_\dagger \leq^\star \vec{w}_\dagger$, which concludes the proof. ◀

[8] This is the only point in which we have to define S^π where S is a *function symbol*, here we use a definition by cases in which every case is a quantifier-free term guarded by a quantifier-free formula.