Traversal-invariant definability and Logarithmic-space computation

Steven Lindell
Haverford College
slindell@haverford.edu

Scott Weinstein
University of Pennsylvania
weinstein@cis.upenn.edu

Abstract

In the presence of arithmetic, order-invariant definability in first-order logic captures constant parallel time, i.e. uniform AC^0 [I]. The ordering is necessary to replicate the presentation of a structure on the input tape of a machine. What if that ordering was in fact a traversal of the structure? We show that an analogous notion of traversal-invariant definability characterizes deterministic logarithmic space (L) and that breadth-first traversals characterize nondeterministic logarithmic space (NL). The surprising feature of these characterizations is that we can work entirely within the framework of first-order logic without any extensions, other than the traversals themselves.

Keywords: first-order logic; logarithmic space computation; order invariant definability

Introduction

In his seminal work on using formulae to describe resource-bounded computation, Neil uses first-order logic (FO) augmented by inductive fixed-point operators (FP) to capture classical complexity classes such as polynomial-time (P), logarithmic-space (L) and non-deterministic logarithmic-space (NL) on finite ordered structures [I]. Without those extensions, Neil showed that FO can still capture constant (parallel) time in the presence of *bit*, the relation which associates elements in numerical order to their binary expansions. Over time, other characterizations of logarithmic space complexity have been proposed, all of which have some mechanism for (limited) recursion, akin to the path operators proposed in Neil's original work. We propose an alternate view of L and NL from the perspective of presentation invariant definability, those queries definable in FO with respect to special representations of their input, and show they are characterized by the most natural of these, traversal invariant definability.[*]

In the descriptive approach to complexity, formulas in extensions of first-order logic used to express the solution to a problem replace algorithms running on resource-bounded models of machine computation. Those extensions have relied on variations of classically developed inductive definability, known as fixed-point logic [M]. For example, a query on finite ordered graphs is expressible in fixed-point logic if and only if it is computable in polynomial-time [I]. Similarly, various transitive-closure logics capture logarithmic-space computability on ordered structures [I].

Order Invariant definability

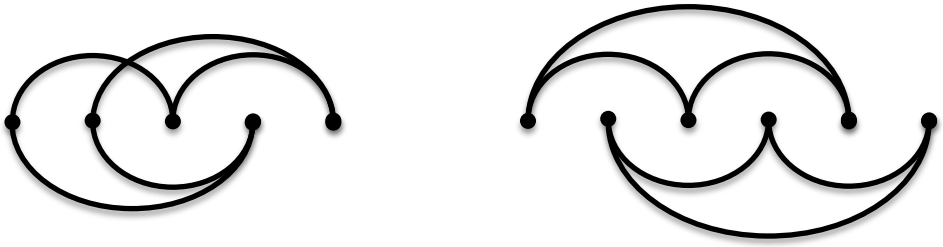
Central to the connection between logic and complexity is understanding the precise role of order, since whenever a structure is given to a machine, it must be presented in some order on the input tape. A machine that purports to compute an isomorphism invariant graph query must output an answer which is independent of that order. Order-invariant first-order logic attempts to capture this notion with an arbitrary order of the input. Although it has greater expressive power, it is still local and so in particular fails to express the connectivity of an undirected graph [L]. While this paper does not address the role of that order in general, it does address the specific role that special traversal orderings play in invariant first-order definability.

Definition: A sentence σ (more generally a formula) is *order-invariant* over finite ordered graphs if for every finite graph G and any two linear orderings $<_1$ and $<_2$ of it, $(G, <_1)$ satisfies σ iff $(G, <_2)$ satisfies σ.

Note: A sentence over ordered graphs is order-invariant in the finite just in case it is invariant under adjacent transpositions, because every finite permutation can be written as the product of adjacent transpositions.

[*] I originally intended to present this at LCC 2014 in Vienna as part of the celebration for Neil Immerman, but was unable to obtain travel funding. I retained the first paragraph of this introduction which still refers to Neil's work.

Order invariant first-order logic on graphs has been studied, and although Yuri Gurevich has shown it is a nontrivial extension of first-order logic, it is well known that connectivity still cannot be defined in this setting. A classic proof also due to Yuri considers the linearly ordered graph in which edges join every other element alternately in the order, wrapping around from highest to lowest as necessary. Formed in this way, odd length orders will consist of a single cycle, and even length orders of two cycles. But a simple compactness argument shows that even and odd length orders cannot be distinguished by any first-order sentence because all infinite models of the theory of finite linear orders are elementarily equivalent.



Left: an odd example, with one cycle. Right: an even example, with two cycles.

Hence, this shows that connectivity is not first-order definable even in the presence of order. In fact, order invariant first-order logic is still local [L]. Moreover, we can extend Yuri's idea slightly by using the same construction on a binary string as a starting point, this time only putting edges between adjacent ones in the order. The resulting graph will be connected if and only if it has odd parity. But we know that parity is not first-order definable, even with arbitrary numerical predicates [FSS], and as already observed there:

Proposition: Connectivity is not first-order definable in the presence of arithmetic [FSS].

But what if instead of assuming our structures are presented in an arbitrary order, we assume they are presented in a particularly natural order? One such notion is that of a traversal order, classically defined for connected graphs [D], easily extended to arbitrary graphs, and even arbitrary structures. All graphs under consideration will be *simple*, i.e. undirected and without loops. Algorithms like breadth-first search yield standard examples of graph traversals. In fact, every finite graph admits many such traversals.

Definition: A *traversal* of a connected graph is a linear ordering of its vertices in which every initial segment is connected. A graph traversal is an arbitrary ordering of its individual component traversals. A traversal of a relational structure S is simply a traversal of its Gaifman graph, the simple graph which connects two distinct elements of S whenever they occur together in some tuple of a relation from S. Notice that the Gaifman graph of a simple graph is just itself.

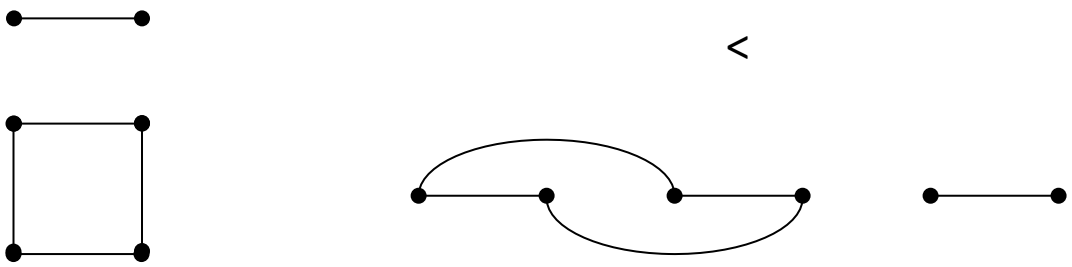


Figure: a graph (left) and one of its traversal presentations (right)

A traversal of a connected graph is characterized by the property that every node has a preceding neighbor. This fact can be used to show that the property of being a traversal is itself elementary [CK].

Traversal invariant definability

Suppose we take first-order formulas on ordered graphs which are invariant of the order, whenever that order is a valid traversal of the graph. In other words, the formula determines a query on graphs given any traversal of the graph.

Definition: A sentence σ (more generally a formula) is *traversal-invariant* over finite ordered graphs if for every finite graph G and any two traversals $<_1$ and $<_2$ of it, $(G, <_1)$ satisfies σ iff $(G, <_2)$ satisfies σ.

Note that the properties of being either connected or acyclic are both traversal-invariant definable. A traversed graph is connected if and only if every element except the first has a preceding neighbor. And a traversed graph is acyclic if and only if it does not have a node with two preceding neighbors.

Even the transitive closure of a graph first-order definable on any graph with a traversal ordering because a smaller node x can reach a larger node y just in case every node in the interval $(x, y]$ has a preceding neighbor. This shows that:

Proposition: Undirected reachability is traversal invariant definable.

Here is a more sophisticated example of expressing a non-local graph query using traversal invariance.

Example: A connected graph G is *bipartite* iff the square of its edge relation is disconnected, because G has an odd cycle if and only if its square is connected. Notice that here we traverse the square of the graph, which is first-order definable from G via $E^2 = E \circ E$.

So, it seems like we should close the idea of traversal invariance under first-order interpretations, leading us to define traversal invariant queries as compositions of: a first-order translation π; a traversal $<$ of (the Gaifman graph G of) the resulting structure; and finally a traversal invariant sentence σ.

Notice that since the traversal of an arbitrary structure is just a traversal of its Gaifman graph, and since the map from an arbitrary structure to its Gaifman graph is first-order definable, we can without loss of generality assume that π produces that simple graph and assume that σ is a formula over ordered simple graphs. In pictures (the traversal is the key middle step):

$$S \to \pi(S) = G \to (G, <) \to \sigma(G, <)$$

Definition: A (Boolean) query is *traversal invariant* if it can be defined by $\sigma(\pi(S), <)$ where π is a first-order interpretation, and σ is a first-order sentence over ordered simple graphs with the property that for any traversal $<$ of $\pi(S)$, whether $(\pi(S), <)$ satisfies σ is independent of the particular traversal $<$ that was chosen. In other words, σ determines a well-defined query using any traversal $<$ of $\pi(S)$ satisfying τ. There is an obvious generalization to non-Boolean queries when σ is a formula which we omit here.

The naturalness of a traversal presentation cannot be overemphasized, because it captures the notion that whenever possible, a new piece of input data is related to some previous piece. One of the most familiar examples of a traversal is breadth-first search. However, it is almost always described procedurally, whereby a method is explained to number and thereby order the vertices according to the BFS algorithm. But a logical presentation helps to reveal its fundamental properties.

Definition: A *breadth-first traversal* is one in which the earliest neighbor function (the parent in the breadth-first tree) is monotone (let the parent of the first element in a component be itself). It is easy to see that this captures the signature queue property of the breadth-first algorithm, since if the vertices are laid out in a line corresponding to their breath-first order, none of the edges in the breadth-first tree will nest, one properly inside the other. Clearly, this property is elementary (first-order definable).

Results

As mentioned earlier, for order invariant first-order logic to capture uniform-AC^0, we need to assume that our logic also has access to arithmetic over that order, which is equivalent to FO(<, *bit*) [I]. Since the bit predicate is nothing more than the set membership relation over hereditarily finite sets in reverse, we define the notion of situated structures analogous to that of ordered structures.

Definition: Call a structure *S situated* if its signature includes an accompanying bit predicate.

These structures are without loss of generality already ordered, since < is elementarily definable from *bit* [DDLW]. We can now state our first result, which characterizes L.

Theorem: A query on situated structures is computable in L iff it is traversal invariant.

Our second result characterizes non-deterministic log-space computability in terms of breadth-first search. That is to say, NL is the class of breadth-first traversal-invariant queries, defined like traversal invariance before except that we are guaranteed the given ordering of the graph is in fact a breadth-first traversal.

Theorem: A query on situated structures is computable in NL iff it is breadth-first invariant.

One might think that including the bit predicate in the initial input is important. But by using the equivalence between logarithmic-space machines and multi-head finite automaton, we can improve both the above results in that we can capture queries on structures that come only with a successor relation and do not require the bit relation. The details of this are quite technically involved and omitted from this abstract.

Extension: A query on structures with successor is in (N)L iff it is (breadth-first) traversal invariant.

Conclusion

We have only examined traversal invariant definability in this paper. However, many other types of ordered presentations (both linear and partial) should be explored. Some of these may allow elementary characterizations of NC^1 or even P which are analogous to the elementary characterizations of logarithmic space computation that we have presented here. In general, presentation invariant definability stays inside of NP ∩ co-NP [GLL].

References

[CK] Corneil, D. & Krueger, R. “A Unified View of Graph Searching” SIAM J. Discrete Math., 22(4), 1259–1276.

[D] Diestel, R. *Graph Theory*, 4th Edition, Springer, 2012.

[DDLW] Dawar, A. & Doets, K. & Lindell, S. & Weinstein, S. “Elementary Properties of the Finite Ranks” Mathematical Logic Quarterly, Vol. 44, pp. 349-353 (1998).

[FSS] Furst, M. & Saxe, J. & Sipser, M. “Parity, Circuits, and the Polynomial-Time Hierarchy” Mathematical Systems Theory, Vol. 17, pp. 13-27 (1984).

[GLL] Grumbach, S. & Lacroix, Z. & Lindell, S. “Generalized Implicit Definitions on Finite Structures” CSL 1995: 252-265

[L] Libkin, L. *Elements of Finite Model Theory* Springer, 2004.

[M] Moschovakis, Y. *Elementary induction on abstract structures* North-Holland, 1974.

[I] Immerman, N. *Descriptive complexity* Springer, 1999.