ELSEVIER

Theoretical Computer Science 148 (1995) 261–279

Theoretical Computer Science

Reachability and the power of local ordering ☆

Kousha Etessami *, Neil Immerman

Computer Science Department, University of Massachusetts, Amherst, MA 01003, USA

Abstract

The L $\stackrel{?}{=}$ NL question remains one of the major unresolved problems in complexity theory. Both L and NL have logical characterizations as the sets of *totally ordered* ($\leqslant$) structures expressible in first-order logic augmented with the appropriate Transitive Closure operator (Immerman, 1987): (FO + DTC + $\leqslant$) captures L and (FO + TC + $\leqslant$) captures NL. On the other hand, in the absence of ordering, (FO + TC) is strictly more powerful than (FO + DTC) (Grädel and McColm, 1992).

An apparently quite different "structured" model of logspace machines is the Jumping Automaton on Graphs (JAG), (Cook and Rockoff, 1980). We show that the JAG model is intimately related to these logics on "one-way locally ordered" (1LO) structures. We argue that the usual JAG model is unreasonably weak and should be replaced, wherever possible, by the two-way JAG model, which we define. Furthermore, the language (FO + DTC + 2LO) over two-way locally ordered (2LO) graphs is more robust than even the two-way JAG model, and yet lower bounds remain accessible. We prove an upper bound on the power of TC over one-way locally ordered graphs, and three lower bounds on DTC.

1. Introduction

The L $\stackrel{?}{=}$ NL question remains one of the major unresolved problems in complexity theory (L = DSPACE[$\log n$] and NL = NSPACE[$\log n$]). Both L and NL have logical characterizations as the sets of *ordered* structures expressible in first-order logic augmented with the appropriate Transitive Closure operator [14]. (FO+DTC+ $\leqslant$) captures L and (FO+TC+ $\leqslant$) captures NL. On the other hand, in the absence of ordering, (FO+TC) is strictly more powerful than (FO+DTC) [12]. (We include a simple proof of this result which applies also to the stronger language (FO + DTC + COUNT) in which counting quantifiers are present.) Attempts to extend this proof to separate the languages with ordering and thus separate L from NL remain unsuccessful.

An apparently quite different "structured" model of logspace machines is the Jumping Automaton on Graphs (JAG) [8]. It is known that the JAG model is not powerful

☆ Both authors were supported by NSF grant CCR-9207797.

* Corresponding author. Email: etessami@cs.umass.edu.

SSDI 0304-3975(95)00034-8

enough to search all graphs. This may be considered as some evidence that L $\neq$ NL. Unfortunately, the same proof shows that the JAG is not powerful enough to search all trees, a problem that is easily seen to be in L. Thus, the JAG model, like the language (FO + DTC) on unordered structures, is too weak to capture deterministic logspace computation.

An interesting feature of the JAG model is that it posits an ordering on the edges leaving each vertex. We call a graph equipped with such an ordering, *one-way locally ordered* (1LO). We show that the language (FO+DTC+1LO) is strictly more powerful than the JAG model: it can do everything the JAG can do; and it can search all trees as well. We also consider the language (FO + TC + 1LO) and we show that in this language a global ordering is definable for all points reachable from a given point. Thus, we have shown that the language (FO + DTC + 1LO) is a more robust version of the JAG model and fits in neatly between L = (FO + DTC + $\leqslant$) and, (FO + DTC).

Looking deeper, we observe that a weakness of the JAG model is that the local ordering only considers outgoing edges: the JAG does not have the power to back up. It is therefore quite interesting to consider the language (FO + DTC + 2LO) with two-way local orderings and its relationship to the analogous class of two-way JAGs. The later model is able to search all trees and is much more robust than the traditional one-way JAG.

Our main lower bounds build on the lower bounds in [8] and in [5]. We show that DTC of first-order formulas in the language (FO + DTC + 1LO) does not suffice to express reachability. We also show that DTC of first-order formulas does not suffice to express reachability for two-way local ordered graphs *without numbers*. A preliminary version of this paper appeared as [9].

2. Descriptive complexity

In this paper our notation follows the conventions of Descriptive Complexity. See [14, 16] for more detail and motivation.

We code all inputs as finite logical structures. The typical example in this paper is a graph,

$$G = \langle \{v_0, v_1, \ldots, v_{n-1}\}, E, s, t \rangle$$

The universe of G, $|G| = \{v_0, v_1, \ldots, v_{n-1}\}$ is the set of vertices and the binary relation E is the edge relation. In this paper our graphs will usually have two specified vertices, $s, t \in |G|$, and we will be particularly interested in the GAP (stCON) problem, i.e., whether there is a path in G from s to t. We will use n to denote the number of vertices of G.

In general, a *vocabulary*

$$\tau = \langle R_1^{a_1}, \ldots, R_t^{a_t}, c_1, \ldots, c_s \rangle$$

is a tuple of input relation symbols and constant symbols. A *structure*

$$\mathcal{A} = \langle \{v_0, v_1, \ldots, v_{n-1}\}, R_1^{\mathcal{A}}, \ldots, R_t^{\mathcal{A}}, c_1^{\mathcal{A}}, \ldots, c_s^{\mathcal{A}} \rangle$$

of vocabulary τ is a finite set $|\mathcal{A}| = \{v_0, v_1, \ldots, v_{n-1}\}$ together with relations $R_i^{\mathcal{A}} \subseteq |\mathcal{A}|^{a_i}$, $i = 1, 2, \ldots, t$, and elements $c_j^{\mathcal{A}}$, $j = 1, 2, \ldots, s$.

Let STRUC[τ] denote the set of all finite structures of vocabulary τ. We define a complexity theoretic *problem* to be any subset of STRUC[τ] for some τ.

For any vocabulary τ there is a corresponding first-order language $\mathcal{L}(\tau)$ built up from the symbols of τ, the logical relation symbol, "=", the logical connectives, $\wedge, \vee, \neg$, variables, and quantifiers, $\forall, \exists$.

A logical structure will be called *ordered* if it includes a relation, $\leqslant$, that represents a total ordering on the universe of the structure. Ordered structures will also have constants $0, \mathbf{m}$ denoting the first and last elements of the universe. We will usually assume for ordered structures that the universe is just $\{0, 1, \ldots, n-1\}$ with the usual ordering.

Let FO be the set of first-order definable problems. FO over ordered structures (with an additional predicate, BIT) is equal to the the low-level complexity class uniform AC^0 which is the set of problems checkable in constant parallel time by CRAMs with polynomial much hardware. A CRAM is a uniform CRCW PRAM.

Fact 2.1 (Immerman [17]). *Over ordered structures with* BIT,

$$\mathrm{FO} = \mathrm{AC}^0 = \mathrm{CRAM}[1]$$

An appealing way to increase the descriptive power of first-order logic, so that it can capture more powerful complexity classes, is by adding various transitive closure operators:

Let $\varphi(x_1, \ldots, x_k, x_1', \ldots, x_k')$ be a formula with the specified $2k$ free variables (φ might also have other free variables). We will write $(\mathrm{TC}_{x_1 \ldots x_k x_1' \ldots x_k'}\ \varphi)$ to denote the reflexive, transitive closure of the binary relation $\varphi(\bar{x}, \bar{x}')$. Let (FO+TC) be the closure of first-order logic with arbitrary occurrences of TC.

Fact 2.2 (Immerman [14, 15]). $(\mathrm{FO} + \mathrm{TC} + \leqslant) = \mathrm{NL}$

A deterministic version of TC called DTC is defined as follows. For any formula $\varphi(\bar{x}, \bar{x}')$ define the deterministic reduct of φ by cutting off all outgoing edges from every vertex that has more than one outgoing edge:

$$\varphi_{\mathrm{d}}(\bar{x}, \bar{x}') \equiv \varphi(\bar{x}, \bar{x}') \ \wedge (\forall \bar{z})(\varphi(\bar{x}, \bar{z}) \ \rightarrow \ \bar{z} = \bar{x}')$$

Define DTC(φ) to be the transitive closure of the deterministic reduct of φ:

$$(\mathrm{DTC}_{x_1 \ldots x_k x_1' \ldots x_k'}\ \varphi) \equiv (\mathrm{TC}_{x_1 \ldots x_k x_1' \ldots x_k'}\ \varphi_d)$$

A formula $\Psi \equiv (\mathrm{DTC}_{x_1 \ldots x_k x_1' \ldots x_k'}\ \psi(\bar{x}, \bar{x}'; \bar{w}))(\bar{u}, \bar{v})$ is somewhat intimidating. We can make it simpler to write by noting that $\bar{x}$ and $\bar{x}'$ are dummy variables and the free variables $\bar{w}, \bar{u}, \bar{v}$ are all just parameters, serving the same role. Observe that, given a vocabulary with constants s and t, we may rewrite Ψ in the form

$$\Psi' \equiv (\mathrm{DTC}_{x_1 \dots x_{k+2} x'_1 \dots x'_{k+2}}\ \psi'(\bar{x}, \bar{x}'; \bar{p}))(\bar{s}, \bar{t})$$

by letting $\bar{p} = \bar{w}, \bar{u}, \bar{v}$ and ψ' be

$$\psi' \equiv (\bar{x} = \bar{s} \wedge\ \bar{x}' = \bar{u}, s, t) \vee (\bar{x} = \bar{v}, s, t \wedge \bar{x}' = \bar{t}) \vee$$
$$(x_{k+1}, x_{k+2} = x'_{k+1}, x'_{k+2} \wedge \varphi(\bar{w}; x_1, \dots, x_k, x'_1, \dots, x'_k)).$$

Ψ can thus be abbreviated as the equivalent and simpler looking formula: $\Psi' \equiv \mathrm{DTC}(\psi'(\bar{p}))$.

Fact 2.3 (Immerman [14]). $(\mathrm{FO} + \mathrm{DTC} + \leqslant) = \mathrm{L}$

Other transitive closure operators are interesting to consider. We briefly mention a third one: Alternating Transitive Closure (ATC) is the generalization of TC to and/or graphs. See [14] for details.

Fact 2.4 (Immerman [14]). $(\mathrm{FO} + \mathrm{ATC} + \leqslant) = \mathrm{P}$

Facts 2.2–2.4 do not go through for nonordered structures. One reason for this is that these languages are overly weak when restricted to graphs without much structure. For example, on a graph with no edges, even (FO+ATC) is powerless to walk through the vertices and so cannot express the proposition that there is an odd number of vertices [13].

On the other hand, the reason we would prefer not adding the ordering relation is that it allows the language to express order-dependent properties, i.e., properties that depend not on the graph but rather on the arbitrary ordering in which the graph is presented. One way to add back some of what is lost when we take away ordering, is to add a second universe of numbers, with an associated ordering. Thus a graph *with numbers* is a two-sorted structure,

$$G = \langle \{0, 1, \dots, n-1\}, \{v_0, v_1, \dots, v_{n-1}\}, \leqslant, 0, \mathbf{m}, E \rangle$$

Here the edge relation E applies to the domain of vertices $\{v_0, v_1, \dots, v_{n-1}\}$ and the ordering $\leqslant$ and constants $0, \mathbf{m}$ refer to the domain of numbers. For convenience we will assume that there are two sorts of variables: number variables $i, j, k, \dots$ and vertex variables $u, v, w, x, y, z, \dots$ *In this paper we will assume that graphs are equipped with numbers, unless we explicitly state otherwise.*

Once numbers are available it is nice to be able to count. To do this, we can add counting quantifiers. Let the meaning of the formula

$$(\exists i\, x)\varphi(x)$$

be that there exist at least i distinct vertices x such that $\varphi(x)$. Note that this quantifier binds x and leaves i free. We will let (FO + COUNT) denote first-order logic over structures with numbers and counting quantifiers. Let ThC^0 be the set of problems checkable by uniform sequences of polynomial size, bounded depth threshold circuits. We have the following relationships: (The relation BIT is mentioned in the following fact. We have avoided talking in detail about BIT because it is definable in (FO + DTC + $\leqslant$) and above [14]; and, thus is not needed in the remainder of this paper.)

Fact 2.5. *For totally ordered structures with BIT the following containments hold. The classes inside boxes are equal. (In this fact we assume that all languages have* $\leqslant$ *and BIT.)*

AC^0	$\subsetneq$	ACC^0	$\subseteq$	ThC^0	$\subseteq$	L	$\subseteq$	NL	$\subseteq$	P
FO		(FO + $\oplus_k$)		(FO + COUNT)		(FO + DTC)		(FO + TC)		(FO + ATC)

Unfortunately, counting does not suffice to replace ordering:

Fact 2.6 (Cai et al. [7]). *There is a property in* (FO+$\oplus_2$+$\leqslant$) *that is not expressible over nonordered structures even in* (FO + COUNT + ATC) *and thus not in* (FO + COUNT + TC) *nor* (FO + COUNT + DTC) *either.*

2.1. Separation of TC logics without ordering

It is known that the language (FO + TC) is more powerful than (FO + DTC) [12]. Here we give a particularly simple proof of this fact and also note that the proof goes through in the presence of counting.

Theorem 2.7. *Reachability from s to t is expressible in* (FO+TC); *but, not in* (FO+DTC), *nor even in* (FO + DTC + COUNT).

Proof. The existence of a path from s to t is expressible in (FO + TC) as follows:

$$\mathrm{TC}_{x,x'}(E(x,x'))(s,t).$$

To prove that this is not expressible in (FO + DTC), we consider the graphs $G_n = (V_n, E_n)$ (see Fig. 1).

$$G_n = (V_n, E_n); \qquad V_n = \{a_i, b_i \mid i = 1,2,\ldots,2n\}$$
$$E_n = \{\langle a_i, a_{i+1}\rangle, \langle a_i, b_{i+1}\rangle, \langle b_i, a_{i+1}\rangle, \langle b_i, b_{i+1}\rangle \mid i = 1,2,\ldots,n-1,n+1,\ldots,2n--1\}.$$

Observe that the transpositions $\pi_i = (a_i\ b_i)$ are automorphisms of G_n for $1 \leqslant i \leqslant 2n$. Thus, for any formula $\varphi(x_1,\ldots,x_k,x'_1,\ldots,x'_k)$ and any pair of k-tuples $\bar{c}, \bar{d}$ from V_n and any i, $1 \leqslant i \leqslant 2n$, we have that

$$G_n \models \varphi(\bar{c},\bar{d}) \quad \Leftrightarrow \quad G_n| \models \varphi(\pi_i(\bar{c}), \pi_i(\bar{d})) \tag{2.1}$$

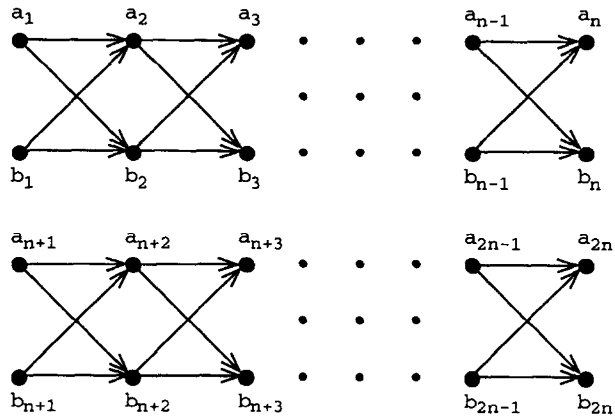

Fig. 1. The graph G_n.

It follows from Eq. 2.1 that if $G_n \models \varphi_d(\bar{c},\bar{d})$ and $\bar{d}$ includes a vertex with subscript i, then so does $\bar{c}$. (Otherwise, $\pi_i(\bar{c}) = \bar{c}$, but $\pi_i(\bar{d}) \neq \bar{d}$ and thus $\bar{c}$ has two outgoing φ-edges.)

It follows that over the graphs $G_n, n = 1,2,\ldots$, DTC(φ) is equivalent to a first-order formula. Thus over these graphs, (FO + DTC) = (FO). Of course reachability is not first-order expressible over the G_n's. (This follows for example from Gaifman's theorem, see Theorem 4.2.)

To show that reachability is not expressible in (FO + COUNT + DTC), we first note that the above argument shows that on the G_n's, (FO + COUNT + DTC) is equal to (FO + COUNT). It now remains to show that (FO + COUNT) cannot express reachability over the G_n's.

The automorphism of the G_n's rendered the DTC operator useless. Now that DTC is gone we lose no generality in considering slightly simpler graphs. Let D_n be the induced subgraph of G_n restricted to the vertices $\{a_1, a_2, \ldots, a_{2n}\}$. In D_n, let $s = a_1$ and $t = a_n$, so that t is reachable from s in D_n. Let D'_n be the same graph but with $t = a_{2n}$, so t is not reachable from s in D'_n. We prove:

Lemma 2.8. *No sentence from* (FO + COUNT) *is true for all the* D_n*'s and false for all the* D'_n*'s.*

Proof. We use the Ehrenfeucht–Fraissé counting game of [7] to prove that D_n and D'_n agree on all sentences from (FO + COUNT) of quantifier-rank $\lfloor \lg(n) - 1 \rfloor$. We must show that Player II – the Duplicator – wins the $\lfloor \lg(n) - 1 \rfloor$-move counting game on D_n and D'_n.

Consider the standard winning strategy for the Duplicator in the game without counting. Namely, for $1 \leqslant i \leqslant n$ the response to the chosen vertex $v \in \{a_i, a_{n+i}\}$ in one graph is $w \in \{a_i, a_{n+i}\}$ in the other graph. The rule for deciding which is that we look at the previously chosen point v_t that is closest to v, and we answer with the w that is on the same line (and thus the same distance) from w_t. If two points on different lines are the same shortest distance away, then we arbitrarily reply with w on the same line as v. An induction argument shows that this is a winning strategy for the Duplicator in the $\lfloor \log n \rfloor - 1$ move game.

Note that the Duplicator's winning strategy gives a 1:1 correspondence between v's and w's. Thus any set chosen by the spoiler in the counting game is answered by a set of the same cardinality. The Duplicator wins the counting game as claimed. □

This completes the proof of Theorem 2.7. □

Note that this theorem is not very satisfying because rather than proving that L $\neq$ NL, it just shows the weakness of the model. It is easy for (FO + DTC) to express reachability over the G_n's in the presence of ordering. Let

$$N(x,x') \equiv E(x,x') \wedge (\forall z)(E(x,z) \rightarrow x' \leqslant z),$$

then

$$\mathrm{PATH}(u,v) \equiv (\exists w)[\mathrm{DTC}(N(x,x'))(u,w) \wedge (v = w \vee E(w,v))]$$

This definition also works in the presence of a one-way (and thus also a two-way) local ordering, cf. Definition 3.1 (4.1).

3. The JAG model: locally ordered graphs

In this section we define the JAG model. We will see that the JAG model is somewhat weaker than (FO + DTC + 1LO), cf. Theorem 3.5. When the JAG is applied to ordered graphs it has the same power as (FO + DTC + $\leqslant$), i.e., exactly L, cf. Proposition 3.6.

As we have indicated, an important issue concerning the power of JAGs is that they take as input one-way locally ordered graphs. We thus first define:

Definition 3.1 (*One-way local ordering*). Consider a graph

$$G = \langle \{0,1,\ldots,n-1\}, \{v_0, v_1, \ldots, v_{n-1}\}, \leqslant, 0, \mathbf{m}, E, F, s, t \rangle$$

in which F is a ternary relation on vertices. Suppose that for each vertex, v, $F(v,\cdot,\cdot)$ is a total ordering on the vertices w for which there is an edge from v to w. Then F is called a *one-way local ordering* on (the outgoing edges of) G, and G is called a *one-way locally ordered* graph. We denote logics over graph structures augmented with one-way local ordering with the abbreviation 1LO.

The following observation gives an alternate way to view local orderings:

Observation 3.2. *In* (FO + DTC + 1LO) *one can express for graphs the relation* $E_i(v,w)$ *meaning that vertex* w *is the head of the* i*th edge out of* v. (*Here* i *is a number variable, not a constant.*)

Proof. Using (FO + DTC + 1LO) we can express the 1:1 correspondence between the numbers $0, 1, \ldots, i$ and the first $i+1$ edges out of v. We first say that a vertex z is the head of the zero$^{\text{th}}$ edge out of v:

$$\zeta(z) \equiv (E(v,z) \wedge (\forall w)(E(v,w) \rightarrow F(v,z,w))$$

Then we say that there is a φ-edge from the pair $\langle j,x\rangle$ to the pair $\langle j+1,y\rangle$ iff y is the head of the next edge out of v after x:

$$\begin{aligned}\varphi(j,x,k,y) \equiv\ & (k = j+1) \wedge x \neq y \wedge F(v,x,y) \wedge (\forall u)[(F(v,x,u) \\ & \wedge F(v,u,y)) \rightarrow (x = u \vee y = u)]\end{aligned}$$

Finally,

$$E_i(v,w) \equiv (\exists z)(\zeta(z) \wedge (\mathrm{DTC}_{jxky}\varphi)(0,z,i,w)). \quad \square$$

We now define the JAG. Note that the JAG defined in [8] is a nonuniform model. We modify the definition here exactly so that the model is uniform:

Definition 3.3 (*JAG*). A uniform *Jumping Automaton on Graphs* (JAG) is a logspace Turing machine that accesses its input via a bounded number of pebbles. Input to a JAG is a one-way locally ordered graph with two specified vertices, s and t. Initially, all the pebbles are on the initial vertex, s. At each move, the JAG can detect which of its pebbles coincide, and which are on s or t. Based on this information, besides making its usual Turing machine moves, it may jump any pebble to the location of another specified pebble, or it may slide a pebble currently at vertex v along a specified edge out of v. Edges are specified by their number in the local ordering $F(v,\cdot,\cdot)$. If there is no such edge, then the pebble remains where it is.

As an example, we prove the following

Proposition 3.4. *The GAP problem for the set of graphs* G_n *of Fig.* 1, *is solvable by a JAG.*

Proof. Define the JAG, J_0, as follows: J_0 needs two pebbles, p_0, p_1, and, does not use its worktape. J_0 begins with its pebbles on vertex s. If $s = t$, then J_0 accepts. Otherwise, at the first move J_0 moves p_0 along the 0 edge out of the current vertex, and p_1 along the 1 edge. If either pebble is on t, J_0 accepts, otherwise it jumps p_1 to p_0, and repeats. J_0 can detect if its current vertex has no outgoing edge because after it tried to slide p_0 along the 0 edge, p_0 and p_1 would still coincide. In this case it should reject. □

We will see in Corollary 3.9 that JAGs are strictly weaker than (FO + DTC + 1LO). Right now we show:

Theorem 3.5. JAG $\subseteq$ (FO + DTC + 1LO)

Proof. Let J be an arbitrary JAG. We must show that there is a sentence $\chi_J \in$ (FO + DTC) such that the set accepted by J is exactly the set of one-way locally ordered graphs that satisfy χ_J.

This is similar to the proof that L $\subseteq$ (FO + DTC + $\leqslant$) [14]. We will use a bounded number of numeric variables to code J's $O(\log n)$ bit worktape. We will use a vertex variable v_i to denote the vertex on which pebble p_i sits. Thus, jumping and coincidence of pebbles is first-order. The movement along edges is expressible in (FO + DTC) by Observation 3.2. Thus, the relation $\mathrm{NEXT}_J(\mathrm{ID}_1,\mathrm{ID}_2)$, meaning that ID_2 follows from ID_1 in one move of J, is expressible. Finally, the acceptance condition is given by

$$\chi_J \equiv \mathrm{DTC}(\mathrm{NEXT}_J)(\mathrm{ID}_0, \mathrm{ID}_f) \qquad \square$$

When a JAG is given an ordered graph we assume that it has a pebble placed on 0 and that it may slide any pebble from vertex i to vertex $i+1$. It is interesting to note that in this case:

Proposition 3.6. *The JAG model over ordered graphs is equivalent to* (FO+DTC+ $\leqslant$), *i.e., it exactly captures* L.

Proof. This can be seen as follows. We use one pebble to simulate each first-order variable. Quantification can be simulated by cycling through all vertices in numeric order. Furthermore, DTC can be simulated by starting at a tuple $\bar{u}$, and cycling through all tuples $\bar{v}$ in lexicographical order. If it is discovered that there is a unique $\bar{v}$ such that $\varphi(\bar{u},\bar{v})$ holds, the JAG shifts the $\bar{u}$ pebbles to $\bar{v}$ and repeats. □

3.1. *Reachability on trees with* DTC

Now we show that in (FO + DTC + 1LO) a total ordering is expressible on trees. It follows from the proof that reachability on trees is expressible as a DTC of a first-order formula. It is interesting to contrast this with Theorem 5.4 which shows that such DTCs cannot express reachability on one-way locally ordered DAGs.

Theorem 3.7. *There is a formula* $\gamma(x,y) \in$ (FO + DTC + 1LO) *which, over connected trees, expresses a linear ordering of the vertices of the tree. Furthermore,* γ *is expressible as a single* DTC *of arity* 2 (*plus booleans*).

Proof. We do a preorder traversal of the tree. The formula π below expresses the next step in this preorder traversal. Then we define $\gamma(x, y)$ to mean that we can get from x to y in this traversal, i.e., x precedes y.

In our definition of π and γ we make use of boolean variables i and j. This is so that the arity of the vertex variables can be kept down to two. In the definition of the preorder traversal, π, when we enter a vertex v for the first time, we will actually enter $(v, 0)$. When we want to leave v for the last time after visiting all of v's children, we will enter the dummy node, $(v, 1)$. Thus the traversal of $V \times \{0\}$ is the preorder traversal, and the vertices $V \times \{1\}$ are just used for bookkeeping.

Let the formula $\sigma(u, v)$ mean that v is u's next sibling:

$$\sigma(u,v) \equiv u \neq v \wedge (\exists p)[F(p,u,v) \wedge (\forall w)(F(p,u,w) \\ \rightarrow (u = w \vee F(p,v,w)))].$$

The preorder traversal, π, and the ordering, γ, are defined as follows:

$$\pi(x,i,x',i') \equiv \delta_0 \vee \delta_1 \vee \delta_2 \vee \delta_3,$$

where

$$\delta_0 \equiv (i = 0 \wedge i' = 0 \wedge E(x,x') \wedge \forall z(\neg\sigma(z,x'))),$$

$$\delta_1 \equiv (i = 0 \wedge i' = 1 \wedge x = x' \wedge (\forall z)\neg E(x,z)),$$

$$\delta_2 \equiv (i = 1 \wedge i' = 0 \wedge \sigma(x,x')),$$

$$\delta_3 \equiv (i = 1 \wedge i' = 1 \wedge E(x',x) \wedge (\forall z)\neg\sigma(x,z)),$$

$$\gamma(a,b) \equiv \mathrm{DTC}(\pi(x,i,x',i'))(a,0,b,0). \qquad \square$$

One can easily modify the above formula for linear ordering to obtain a formula for s-t-reachability, just by conjoining π with $\neg(x = s \wedge i = 1)$. This guarantees that π does not ascend from s, and thus that $\gamma(s,t)$ holds iff t is a descendant of s.

In fact, reachability on directed trees is expressible in (FO + DTC) without local ordering. This is because each vertex besides the root has a unique edge to it. Thus we can use DTC to walk backwards. Thus the following formula expresses reachability for directed trees:

$$\mathrm{Reach}(s,t) \equiv \mathrm{DTC}_{xx'}(E(x',x))(t,s)$$

This proves:

Proposition 3.8. *Reachability on trees (even without local ordering) is expressible in* (FO + DTC). *In fact, it is expressible as a single DTC of arity* 2.

Cook and Rackoff proved that reachability on trees is not checkable by JAGS (see Fact 5.3). Together with Theorem 3.5 and Proposition 3.8, this yields:

Corollary 3.9. $\mathrm{JAG} \subsetneq (\mathrm{FO} + \mathrm{DTC} + 1\mathrm{LO})$

3.2. *One-way local ordering and* (FO + TC)

The empty set is a local ordering for any graph with no edges. It follows that:

Proposition 3.10. *Having an odd number of vertices is not expressible in* (FO + TC + 1LO).

However, the more interesting situation is when the graph is connected:

Theorem 3.11. *There is a formula* $\lambda(x,y) \in$ (FO + TC + 1LO) *that describes a total ordering on the vertices reachable from* s.

Proof. We only consider those vertices reachable from s. We first construct a formula $\delta(x,y,i)$ which means that the distance from x to y is i. This is done below as follows: σ identifies a single step; $\mu(u,v,i)$ takes the transitive closure of σ, asserting that there is a path from u to v of length i; and then δ is defined using μ:

$$\begin{aligned}
&\sigma(w,j,w',j') \equiv (j' = j+1) \wedge E(w,w'),\\
&\quad \mu(u,v,i) \equiv \mathrm{TC}(\sigma)(u,0,v,i),\\
&\quad \delta(x,y,i) \equiv \mu(x,y,i) \wedge (\forall k < i)(\neg\mu(x,y,k)).
\end{aligned}$$

Next, we construct a formula $\alpha(z,x)$ which means that z occurs on the lexicographically first, shortest path from s to x. This is done again by taking single steps: $\rho(u,i,u',i')$ means that if u occurs on the lexicographically first, shortest path from s to x and is distance i from x, then u' is the next vertex on this path and is distance $i' = i-1$ from x:

$$\begin{aligned}
\rho(u,i,u',i') \equiv (i' = i-1 \wedge E(u,u')\\
\wedge\, \delta(u',x,i') \wedge (\forall v)[(F(u,v,u') \wedge \delta(v,x,i')) \rightarrow v = u'])
\end{aligned}$$

and α is a transitive closure of ρ:

$$\alpha(z,x) \equiv (\exists d)(\delta(s,x,d) \wedge (\exists i)\mathrm{TC}(\rho)(s,d,z,i))$$

Now, define the total ordering $\lambda(x,y)$ to mean that the distance from s to x is less than the distance from s to y, or the distances are equal, but the lexicographically first, shortest path from s to x precedes the lexicographically first, shortest path from s to y:

$$\begin{aligned}
\lambda(x,y) \equiv (\exists ij)(\delta(s,x,i) \wedge \delta(s,y,j) \wedge (i < j(i = j \wedge (\exists zuvk)(\alpha(z,x) \wedge \alpha(z,y)\\
\wedge \alpha(u,x) \wedge \alpha(v,y) \wedge (F(z,u,v) \wedge u \neq v))))
\end{aligned}$$

In the above, z is the last vertex on which those two lexicographically first, shortest paths agree. $\square$

Theorem 3.11 shows that for the interesting case, ordering is definable in (FO + TC + 1LO). In [9] we conjectured that (FO + TC + COUNT + 1LO) would capture all of NL. This turns out to be false. However, (FO + TC + COUNT + 1LO) = NL. Both of these facts appear in [10] and are discussed further in Section 7.

4. Two-way locally ordered graphs

The feature of the JAG model that makes it unrealistically weak is its inability to back up. This is the reason why JAGs cannot search trees as they should be able to, unlike the language (FO + DTC) which can: Proposition 3.8.

In (FO + DTC) we can usually back up. Namely, if we are at a vertex a that has an edge coming into it from vertex b and vertex b has some special property (such as being the only vertex x such that the 17th edge out of x is to a) then we can back up to b. On the other hand, we can construct our graphs so that all vertices of interest have duplicate, "shadow" predecessors that look locally identical to each real predecessor. In this way we can force the language (FO+DTC+1LO) to be artificially weak. We will exploit this idea to get a general lower bound for (FO + DTC + 1LO): Theorem 5.4.

For this reason, we feel that it is more reasonable to consider graphs equipped with a two-way local ordering:

Definition 4.1. A *two-way local ordering* (denoted by 2LO) is just a one-way local ordering, H, on the incoming edges to each vertex, in addition to the one-way local ordering, F, on the outgoing edges. There is no assumption about consistency between F and H.

A *two-way JAG* is a JAG that takes as input graphs with a two-way local ordering. At each move, the JAG may choose a specified incoming or outgoing edge and move a pebble along it. From now on, we will refer to the usual JAG model as a *one-way JAG.*

4.1. (FO + DTC) and two-way JAGs

Now we show that the language DTC(FO + 2LO) – the restriction of (FO + DTC + 2LO) to single DTCs of first-order formulas – is essentially equivalent to the two-way JAG. The first observation we make is the well-known fact that fixed first-order statements are essentially local in nature. This is stated nicely, for example, in the following theorem of Gaifman. Any logical structure $\mathcal{A}$ may be thought of as a generalized graph in which an edge exists between two points a and b iff a and b occur together in some tuple of a relation in $\mathcal{A}$. The distance between a and b (dist(a,b)) is the minimum number of such edges that must be traversed to get from a to b.

If we restrict all quantifiers in a formula φ to the union of the balls of distance d around a set of points, $\bar{v}$, then we get a local formula, denoted $\varphi^d(\bar{v})$. Gaifman's theorem says that a first-order formula may describe the local neighborhood around its free variables and constants (1), and it may describe the existence of certain landmarks (2), and that is all:

Fact 4.2 (Gaifman [11]). *Let φ be a first-order formula whose free variables and constants are in the tuple $\bar{u}$. Then there exists a distance d depending only on φ such that φ is equivalent to a boolean combination of*

1. *a finite set of formulas $\alpha_i^d(\bar{u})$, and,*
2. *a finite set of sentences (in which no constants nor free variables occur),*

$$\beta_j \equiv (\exists v_1, \ldots, v_s)\left[\bigwedge_{i=1}^{s} \psi^d(v_i) \wedge \bigwedge_{i<j\leqslant s} \mathrm{dist}(v_i, v_j) > 2d\right]$$

We now use Fact 4.2 to analyze the computation performed by the expression $\mathrm{DTC}[\varphi(\bar{x},\bar{x}')](\bar{c},\bar{e})$, where φ is an arbitrary first-order formula.

We know that φ is equivalent to a boolean combination of some β_j's asserting the existence of some distant landmarks; and, some α_i's which are local facts about $\bar{x},\bar{x}'$, and the only constants available: s and t. We may assume that all of the β_j's are satisfiable because any unsatisfiable ones are just superfluous. It follows that there is a one-way locally ordered graph L_φ that contains neither s nor t, but does satisfy all of the β_j's; i.e., L_φ contains all the relevant landmarks. By only considering graphs that include a copy of L_φ disjoint from everything else, we reduce φ to a boolean combination of α_i's which is thus equivalent to one fixed formula,

$$\varphi(\bar{x},\bar{x}') \equiv \alpha^d(\bar{x},\bar{x}',s,t)$$

Now, consider a $\mathrm{DTC}(\varphi)$ walk. Call the step from $\bar{a}$ to $\bar{b}$ *d-local* (or just *local* if d is understood) if every point b_i in $\bar{b}$ is within the d-neighborhood of some point in $\bar{a} \cup \{s,t\}$. If the step is not local because of b_i, then the point b_i must have been the unique point in the graph with some special property. Suppose now, that we add two new, disjoint copies of the graph (with the vertices s and t not labelled in the new copies). Then in this expanded graph there are two equally valid points b_i that we could move to. Thus, in the expanded graph all steps are local.

We summarize the above discussion in the following definition, observation, and lemma:

Definition 4.3. Let φ be a fixed first-order formula as in the above discussion. Let $\bar{p}$ be the set of parameters in φ, i.e., constants s and t plus any other constants or assigned variables. We will call a graph G *adequate for* φ (or just *adequate* if φ is understood) iff G satisfies all the β_i's of Fact 4.2; and, for every point $a \in V_G$, such that $\mathrm{dist}(a,\bar{p}) > d$, G contains another point b where $\mathrm{dist}(b,\bar{p}) > d$ and $\mathrm{dist}(a,b) > 2d$ and b's d-local neighborhood is isomorphic to a's.

The following observation shows that without loss of much generality we can restrict our attention to adequate graphs.

Observation 4.4. *Let G be any graph containing the constants s and t. Given φ, a formula with v variables, let* adq(G) *be the disjoint union of* $2v+1$ *one-way locally ordered graphs: $G, G_1, \ldots, G_v, H_1, \ldots, H_v$ where $G \cong G_1 \cong \cdots \cong G_v$, and $H_1 \cong \cdots \cong H_v \cong L_\varphi$. Then* adq$(G)$ *is adequate for φ. Furthermore, in terms of reachability from s to t and as inputs to JAG-like automata, G and* adq(G) *are indistinguishable.*

Lemma 4.5. *Let $\varphi(\bar{x}, \bar{x}'; \bar{p})$ be a first-order formula where $\bar{p}$ is the set of parameters, i.e., constants and free variables besides $\bar{x}, \bar{y}$. Then there exists a constant d depending only on φ and there exists a d-local formula $\alpha^d(\bar{x}, \bar{x}', \bar{p})$ such that for every graph G that is adequate for φ, we have:*

1. *In G, α is equivalent to φ.*
2. *Every step in every* DTC(φ) *walk in G is d-local.*

A consequence of Lemma 4.5 is that the two-way JAG is very similar to the language (FO + DTC + 2LO). In particular, a lower bound in one of these models translates to a lower bound in the other:

Theorem 4.6. *Let $\mathcal{G}$ be a class of two-way locally ordered graphs with numbers that is closed under the* adq *operation of Observation* 4.4. *Then the following two statements are equivalent:*

1. *For some first-order formula φ,* DTC$[\varphi(\bar{x}, \bar{x}')](\bar{s}, \bar{t})$*expresses reachability from s to t for $\mathcal{G}$.*
2. *There exists a two-way JAG that recognizes reachability from s to t for $\mathcal{G}$.*

On the other hand, if $\mathcal{G}$ is as above but without numbers, then condition (1) *is equivalent to*

3. *There exists a finite state, two-way JAG that recognizes reachability from s to t for $\mathcal{G}$.*

Proof. One direction of this follows from Theorem 3.5. The other direction follows from Lemma 4.5. The JAG can simulate the DTC because it can exhaustively visit all vertices of distance at most d and thus choose the correct tuple to go to next. If we remove numbers and the JAG's worktape then the simulations go through and we have that 1 and 3 are equivalent. Note that since d is a fixed constant, a finite state JAG can do the simulation. □

5. Lower bound: (FO + DTC + 1LO)

In this and the following section, we use Theorem 4.6, together with two previous lower bounds, to prove lower bounds on the descriptive power of (FO + DTC + 1LO), and then in the next section on (FO + DTC + 2LO).

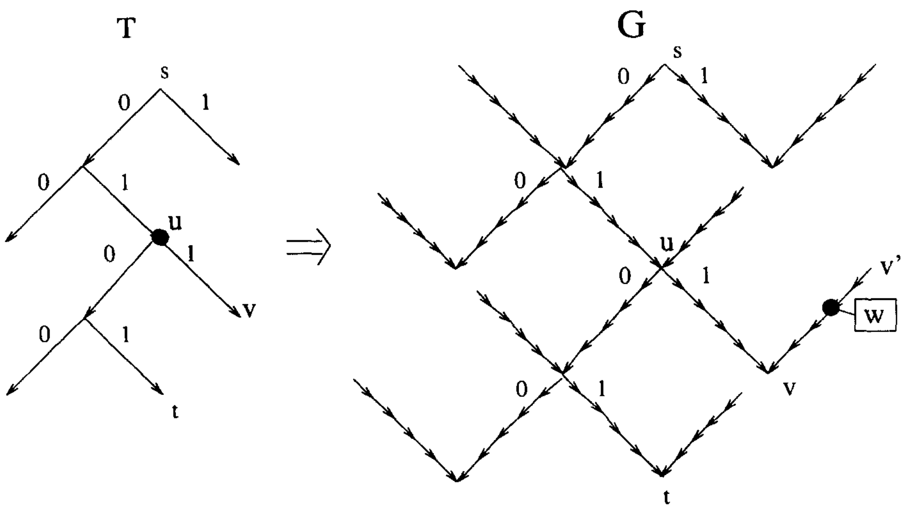

Fig. 2. $G = shadow(T, \delta)$

As mentioned at the beginning of Section 4, a weakness of one-way JAGs is that they cannot back up.We can however similarly prevent DTC walks from backing up by replacing a tree T by a *shadowed* counterpart T', where for each edge $\langle v, w\rangle$ we introduce a new vertex v' and edge $\langle v', w\rangle$ that will be locally indistinguishable from $\langle v, w\rangle$.

Definition 5.1. Given a directed forest T, and a number δ, we define a new DAG G called the shadowed forest:

$$G \leftarrow shadow(T, \delta)$$

G is constructed as follows: First, for each edge (u, v) of T create a new vertex v' and an edge (v', v). Next, replace each old and new edge by a path of length δ,keeping the original local ordering for vertices with out-degree > 1. We call the vertices in G corresponding to the original vertices in the forest tree vertices. Every labelled vertex in T will maintain the same label in G.

For a given vertex w in G, the tree vertex associated with w, denoted $\mathrm{tv}(w)$, is the vertex in the original forest T immediately "above" w. If w is on a shadow path (v', v), then "above" is taken to mean the tree vertex u at the top of the shadowed path (see Fig. 2).

Lemma 5.2. *For any first-order formula φ and DTC formula $\Phi = [\mathrm{DTC}\ \lambda\bar{x}\bar{x'}\ (\varphi)]$, there is a logspace JAG J, and distance d, such that J simulates φ in the following sense. For any forest F, let $F' = adq(shadow(F, d))$. Then*

$$(J\ accepts\ F) \quad \Leftrightarrow \quad (F' \models \Phi)$$

Proof. By Lemma 4.5 we may assume that there is a constant d such that φ is d-local and every φ step is d-local. Let $\delta = 3d + 1$. Furthermore, we may assume that φ is deterministic, i.e., we have already cut off any double outgoing edges. Using the number variables, we can also insure that φ keeps a counter to check itself for looping and that it has no outgoing edges after its counter overflows.

We build J so that it simulates Φ step by step. Initially, J has one pebble on t and the rest of its pebbles on s. At each step, J will start with its x_i pebble on $\mathrm{tv}(x_i)$, the vertex in F that is "above" the vertex $x_i \in F'$. J will also keep on its worktape the distance d_i from $\mathrm{tv}(x_i)$ to x_i in F'. For those x_i that are numeric variables, J will just keep this numeric value on its worktape.

At each step, J must find the tuple $\overline{x'}$ of points in F' such that

$$F' \models \varphi(\bar{x}, \overline{x'}) \tag{5.1}$$

if it exists. Observe that if such a tuple $\overline{x'}$ exists, then each corresponding vertex $\mathrm{tv}(x_i')$ is equal to a currently pebbled vertex, or is an immediate descendent of a currently pebbled vertex. This is the main point of the construction. The reason is that (1) φ cannot talk about distances greater than d and so cannot jump a whole edge from F; and, (2) if there are no current points above a node $v \in F$, then the path from u to v is indistinguishable by φ from the shadow path from u' to v. Thus, if there were an x_i' on one of these paths, then the corresponding x_i' on the other path would also satisfy φ and, thus, there would be no outgoing φ-path.

Thus, to test for a possible $\overline{x'}$, all that J has to do is cycle through all the bounded number of possible movements of its $\overline{x'}$ pebbles by leaving them where they are, jumping them to another pebble, or sliding them along a single edge. For each such placement of the pebbles, J must cycle through all the possible distances from them that are less than or equal to d.

Finally, for each candidate tuple, J must test whether Eq. (5.1) holds. Again, since φ is d-local, this amounts to cycling through all the nearby values of the variables quantified in φ. Note, that in this case quantified vertices may assume positions that are slightly above a pebbled vertex. However, J knows that such vertices exist. (Unless the vertex is s in which case J would know that they do not exist. We are assuming that t is never a root and thus s is the only root that J can reach. Note that to check the truth of Eq. (5.1), J will employ two auxiliary pebbles which it can slide one step down from newly pebbled vertices to check for leaves.) Thus J can keep track of all possible assignments to these variables long enough to test Eq. (5.1).

Thus J simulates Φ and accepts iff it reaches a situation where $\overline{x'} = \bar{t}$. □

To prove our lower bound on (FO + DTC + 1LO) we use the following fact from [8].

Fact 5.3 (Cook and Rackoff [8]). *No JAG can decide s-t-reachability on forests.*

Theorem 5.4. *No formula* $\Phi = [\mathrm{DTC}\ \lambda\bar{x},\bar{x}'\ (\varphi)]$ *expresses s–t-reachability on one-way locally ordered DAGs.*

6. Lower bound: (FO + DTC + 2LO)

It is harder to prove a lower bound on two-way locally ordered graphs. In particular, reachability on the above shadowed trees equipped with a two-way local ordering is expressible as a single DTC.

Two known lower bounds on undirected graphs, together with Theorem 4.6, give a lower bound on the language (FO + DTC + 2LO) without numbers. Recall from Theorem 4.6 that (FO + DTC) without numbers corresponds to finite state JAGs as opposed to logtape JAGs.

Blum and Sakoda considered finite automata walking on three-dimensional mazes. These mazes are subsets of $\mathbf{Z}^n$, where each node has some subset of its six possible neighbors consistently marked north, south, east, west, up, and down.

Fact 6.1 (Blum and Sakoda [5]). *No bounded set of finite automata can search all three-dimensional mazes.*

The following fact by Cook and Rackoff proves a somewhat stronger result on a slightly weaker model. A finite state JAG is stronger than a bounded set of finite automata in that the automata cannot communicate unless they run into each other. Another way to think of a finite state JAG is that it is a bounded set of finite automata equipped with walkie talkies so that they can ask each other what state they are currently in. The following lower bound involves locally consistent two-way local orderings, but the graphs in question curve around themselves rather than being embeddable in $\mathbf{Z}^3$. See also [6] for a related lower bound.

Fact 6.2 (Cook and Rackoff [8]). *No finite state JAG can search all degree three undirected graphs.*

Corollary 6.3. *The reachability problem for undirected graphs equipped with two-way local orderings is not expressible in the form* $\mathrm{DTC}(\varphi)$, *with* φ *first-order and without numbers. Furthermore, the same problem for mazes embedded in* $\mathbf{Z}^3$ *with the inherited directions north, south, east, west, up, and down is not expressible as an arity two DTC.*

7. Conclusion

We have begun an investigation of transitive closure logics applied to locally ordered graphs. We have shown that the JAG model is intimately related to these logics. We have indicated why the JAG model is unreasonably weak and should, wherever possible, be replaced by the two-way JAG model. Furthermore, we have shown that the language (FO + DTC + 2LO) is more robust than even the two-way JAG model, and yet lower bounds remain accessible.

We have proved an interesting upper bound on the power of TC over one-way locally ordered graphs (Theorem 3.11), and three lower bounds on DTC (Theorems 2.7 and 5.4, and Corollary 6.3).

We hope that we have given convincing evidence that a further study of the relationship between (FO + DTC) and (FO + TC) is both feasible and important for understanding the relationship between L and NL.

The following topics merit further study:

1. The lower bounds of Theorem 5.4 and Corollary 6.3 prove the impossibility of expressing reachability by formulas of the form $\mathrm{DTC}[\varphi](\bar{s},\bar{t})$, where φ is first-order. We conjecture that the lower bound of Theorem 5.4 holds for all of (FO+DTC+1LO). There is a normal form theorem which says that every formula in (FO+DTC+ $\leqslant$) can be written in the form $\mathrm{DTC}[\varphi](\bar{s},\bar{t})$ where φ is not only first-order, but quantifier-free [14]. This normal form theorem is false without ordering, even with two-way local ordering. However, we feel that a generalization of the proof of Theorem 5.4 will extend to a lower bound on all of (FO + DTC+1LO).

2. We have argued that (FO + COUNT + DTC + 2LO) is a robust approximation to L and yet admits tractable approaches to lower bounds. Much further study is needed. In particular, lower bounds on (FO + COUNT + DTC + 2LO) are very desirable. Such a lower bound would at least show us a deficiency of this language which we could then fix. At best, such a lower bound could prove that L $\neq$ NL.

3. Related to 2 is the following challenge: find a set of graphs for which reachability is in L but not in (FO + DTC + COUNT + 2LO).

As mentioned in Section 3.2, we had in the conference version of this paper conjectured that the language (FO+TC+COUNT+1LO) would capture all of NL. What remained to be shown to prove this conjecture was that the language (FO+TC+COUNT) canonizes all trees, cf. [18, 19]. However, we have recently [10] proven that tree canonization is not in (FO + TC + COUNT) and, using the result of [19], this yields that (FO + TC + COUNT + 1LO) $\neq$ NL. However, in that same paper we prove that (FO + TC + COUNT + 2LO) = NL, further justifying the continued study of local orderings.

Acknowledgements

Thanks to Allan Borodin and Dexter Kozen for pointing out Facts 6.2 and 6.1 to us.

References

[1] M. Ajtai and R. Fagin, Reachability is harder for directed than for undirected graphs, *J. Symbolic Logic* **55** (1990) 113–150.

[2] R. Aleliunas, R. Karp, R. Lipton, L. Lovasz and C. Rackoff, Random walks, universal traversal sequences, and the complexity of the maze problem, in: *Proc. 20th Ann. IEEE Found. of Comput. Sci. Symp.* (1979) 218–223.

[3] P. Beame, A. Borodin, P. Raghavan, W. Ruzzo and M. Tompa, Time-space tradeoffs for undirected graph traversal, in: *Proc. 31st Ann. IEEE Found. of Comput. Sci. Symp.* (1990) 429–438.
[4] M. Blum and D. Kozen, On the power of the compass, in: *Proc. 19th Ann. IEEE Found. of Comput. Sci. Symp.* (1978) 132–142.
[5] M. Blum and W.J. Sakoda, On the capability of finite automata in 2 and 3 dimensional space, in: *Proc. 18th Ann. IEEE Found. of Comput. Sci. Symp.* (1977) 147–161.
[6] M. Bull and A. Hemmerling, Traps for jumping multihead counter automata, *J. Inform. Process. Cybernet.* **EIK 28** (1992) 343–361.
[7] J. Cai, M. Fürer and N. Immerman, An optimal lower bound on the number of variables for graph identification, *Combinatorica* **12** (1992) 389–410.
[8] S.A. Cook and C.W. Rackoff, Space lower bounds for maze threadability of restricted machines, *SIAM J. Comput.* **9** (1980) 636–652.
[9] K. Etessami and N. Immerman, Reachability and the power of local ordering, in: *Proc. 11th Ann. Symp. Theoretical Aspects Comput. Sci.* (1994) 123–135.
[10] K. Etessami and N. Immerman, Tree canonization and transtive closure, in: *Proc. 10th Ann. IEEE Symp. on Logic in Comput. Sci.*, to appear.
[11] H. Gaifman, On local and non-local properties, in: *Proc. Herbrand Logic Colloq.* (1981) 105–135.
[12] E. Grädel and G. McColm, Deterministic vs. nondeterministic transitive closure logic, in: *Proc. 7th IEEE Conf. on Logic in Computer Science* (1992).
[13] N. Immerman, Upper and lower bounds for first order expressibility, *J. Comput. System Sci.* **25** (1982) 76–98.
[14] N. Immerman, Languages that capture complexity classes, *SIAM J. Comput.* **16** (1987) 760–778.
[15] N. Immerman, Nondeterministic space is closed under complementation, *SIAM J. Comput.* **17** (1988) 935–938.
[16] N. Immerman, Descriptive and computational complexity, in: J. Hartmanis, ed., *Computational Complexity Theory, Proc. Symp. in Applied Math.*, Vol. 38 (Amer. Mathematical Soc., Providence, RI, 1989) 75–91.
[17] N. Immerman, Expressibility and parallel complexity, *SIAM J. Comput* **18** (1989) 625–638.
[18] N. Immerman and E. Lander, Describing graphs: a first-order approach to graph canonization, in: A. Selman, ed., *Complexity Theory Retrospective* (Springer, Berlin 1990) 59–81.
[19] S. Lindell, A logspace algorithm for tree canonization, *ACM Symp. Theory Comput.* (1992) 400–404.