

Space Complexity of Reachability Testing in Labelled Graphs

Vidhya Ramaswamy, Jayalal Sarma, and K.S. Sunil^(✉)

Indian Institute of Technology Madras, Chennai, India
vidhya0794@gmail.com, jayalal.sarma@gmail.com, sunks@gmail.com

Abstract. Fix an algebraic structure $(\mathcal{A}, *)$. Given a graph $G = (V, E)$ and the labelling function ϕ ($\phi : E \rightarrow \mathcal{A}$) for the edges, two nodes $s, t \in V$, and a subset $F \subseteq \mathcal{A}$, the \mathcal{A} -REACH problem asks if there is a path p (need not be simple) from s to t whose yield (result of the operation in the ordered set of the labels of the edges constituting the path) is in F . On the complexity frontier of this problem, we show the following results.

- When \mathcal{A} is a group whose size is polynomially bounded in the size of the graph (hence equivalently presented as a multiplication table at the input), and the graph is undirected, the \mathcal{A} -REACH problem is in L. Building on this, using a decomposition in [4], we show that, when \mathcal{A} is a fixed quasi-group, and the graph is undirected, the \mathcal{A} -REACH problem is in L. In contrast, we show NL-hardness of the problem over bidirected graphs, when \mathcal{A} is a matrix group over \mathbb{Q} . When \mathcal{A} is a fixed aperiodic monoid, we show that the problem is NL-complete.
- As our main theorem, we prove a dichotomy for graphs labelled with fixed aperiodic monoids by showing that for every fixed aperiodic monoid \mathcal{A} , \mathcal{A} -REACH problem is either in L or is NL-complete.
- We show that there exists a monoid M , such that the reachability problem in general DAGs can be reduced to \mathcal{A} -REACH problem for planar non-bipartite DAGs labelled with M . In contrast, we show that if the planar DAGs that we obtain above are bipartite, the problem can be further reduced to reachability testing in planar DAGs and hence is in UL.

1 Introduction

The reachability problem on combinatorial structures has been fundamental and well studied in complexity theory. A most striking example of this is the graph reachability problem which asks, given a directed graph G and two special vertices s and t whether there is a path from s to t in G or not. The problem is known to be NL-complete for directed acyclic graphs. Deterministic logspace algorithms are known for restricted classes of graphs - when each component of the directed graph is Eulerian [19], or the graph has bounded treewidth [10]. Reachability for planar graphs is in unambiguous¹ logspace [7]. See [1] for a survey.

¹ A language is said to be in unambiguous logspace if there exists a non-deterministic logspace Turing machine M such that $\forall x$, M has at most one accepting computation.

Word problems over algebraic structures also play a fundamental role in complexity theoretic characterizations. Fix an algebraic structure \mathcal{A} with an associated binary operation, and a subset $F \subseteq \mathcal{A}$. Given a $w \in \mathcal{A}^*$, test if the sequences of elements and operations among them is in F or not. An important milestone in this direction is the dichotomy result due to Barrington and Thérien [3] classifying the complexity of the word problems over a fixed monoid structure: if the monoid M contains at least one non-solvable group, then the word problem can be shown to be complete for NC^1 (under AC^0 projections) and if all groups are solvable then it characterizes ACC^0 . Chandra *et al.* [9] showed that if there are no non-trivial groups then it characterizes the class AC^0 . It is also known that word problems over groupoids characterize LogCFL [6]. Beaudry *et al.* [5] showed dichotomy theorem for the complexity of circuit evaluation problem defined over monoids, based on precise algebraic properties of the monoids.

Reachability on labelled graphs is a natural generalization of the graph reachability problem and the word problem on algebraic structures. A graph G is said to be *labelled* if the edges are assigned labels from an underlying set S . When this set also equipped a binary operation $* : S \times S \rightarrow S$, the reachability problem asks to test, given the graph G and two vertices s and t and an element $a \in S$, if there is a path (need not be simple) from s to t whose yield (result of the operation in the ordered set of the labels of the edges constituting the path) is a or not. A closely related problem is that of the L-REACH problem where a language L over the alphabet Σ , given a graph $G(V, E)$, two vertices s and t and a labelling function $\phi : E \rightarrow \Sigma$, test if there is a path from s to t whose yield (the concatenation of the labels in the ordered set of edges constituting the path) belongs to the language L . In [15], characterizations of the language reachability problem with respect to languages classes and graph classes were obtained.

In this work, we study the problem when the labels come from richer algebraic structures. When the structure \mathcal{A} is a groupoid (equivalently a case of L-REACH problem when the language is restricted to be a context-free language) this problem has been used in inter-procedural slicing and inter-procedural data flow analysis [12, 20, 21]. On the complexity frontier, it is easy to observe that the \mathcal{A} -REACH problem is always harder than the word problem over \mathcal{A} , and is harder than the graph reachability problem (under logspace many-one reductions).

Our Results: We start with an observation that the problem of testing reachability on labelled graphs over semigroups can be reduced to testing reachability on an associated directed graph (called product graphs - see Sect. 3). From a complexity-theoretic view point, this motivates the study of the labelled reachability problem. More specifically, in order to show that for a graph class the reachability problem is in L, it is sufficient to reduce it to the reachability problem in a labelled graph such that reachability in the product graph can be solved in L. We prove several properties of this directed graph and explore graph classes and algebraic structures for which the \mathcal{A} -REACH problem is in L. In particular, we study this for undirected graphs (for which reachability problem is in L [18]) and show:

Logspace Upper Bounds for Polynomially Growing Groups: We show that when \mathcal{A} is a group, such that $|\mathcal{A}| = O(n^c)$, where n is the size of the graph, and c is a constant (hence equivalently presented as a multiplication table at the input), and the graph is undirected, the \mathcal{A} -REACH problem is L-complete.

NL-hardness for Monoids and Matrix Groups: In contrast, we observe that there exists a fixed monoid \mathcal{A} such that \mathcal{A} -REACH problem for undirected graphs is NL-complete. Working over a more structured labelling set, we show NL-hardness of the problem over bidirected graphs², when \mathcal{A} is a finitely generated subgroup of $\text{GL}_k(\mathbb{Q})$ (for $k \geq 2$)- the group of invertible $k \times k$ matrices with rational entries.

(NL vs L) Dichotomy for Aperiodic Monoids: We show a dichotomy for aperiodic monoids: for any fixed aperiodic monoid \mathcal{A} , the \mathcal{A} -REACH problem for undirected graphs is either NL-complete or is in L.

Logspace Upper Bounds for Quasigroups: When \mathcal{A} is a fixed quasigroup, the \mathcal{A} -REACH problem is L-complete.

Logspace Upper Bounds for Treewidth k Graphs labelled with Monoids: When the graph has bounded treewidth, for any fixed monoid \mathcal{A} the \mathcal{A} -REACH problem for undirected graphs is L-complete.

While the general DAGREACH problem is complete for NL, the reachability problem over planar DAGs is known to be in UL [7]. We show that DAGREACH can be reduced to \mathcal{A} -REACH over planar DAGs when \mathcal{A} is a specific monoid M . Tightly complementing this, we show that the instances of labelled graph reachability problem obtained in this reduction, with the additional restriction that the graph is bipartite, can be solved in deterministic log-space. Moving towards groups, we show that DAGREACH can be reduced to \mathcal{A} -REACH over planar graph when \mathcal{A} is a specific exponentially growing group.

Related Work: Group labelled graphs have been extensively studied in the literature as a generalization of signed graphs (see [11]) with the aim to extend the graph minor theory to group labelled graphs over a fixed finite Abelian group. An important comparison that we make is with the results by Kawase *et al.* [14], where they consider the reachability problem in group labelled graphs: to check if there is a *simple path* from s to t in the given group labelled graph so that the yield of the path is a given element α . It is observed in [14] that the problem is NP-complete over \mathbb{Z} , since the undirected Hamiltonian path problem reduces to this problem by replacing each edge with a pair of two arcs of opposite directions with label 1 and letting $\alpha = n - 1$. Huynh [13] showed the problem is polynomial time solvable if the group is a fixed Abelian group.

However, we point out two important differences in our setting. Firstly, in our setting, the problem does not look for simple paths from s to t , and hence

² Directed graph $(G(V, E))$ such that $\forall v_i, v_j \in V, (v_i, v_j) \in E \implies (v_j, v_i) \in E$. However, $\phi(v_i, v_j)$ need not be equal to $\phi(v_j, v_i)$. To complement this, we observe (see Corollary 1) that the log-space upper bound for groups whose size is polynomially bounded in terms of input, holds even for bidirected graphs.

the NP-completeness result does not apply. Secondly, for the case of undirected graphs with labelling from a group, the above mentioned results (including [14]) assume that if an edge (u, v) in the graph is labelled with g , the edge (v, u) is labelled with g^{-1} . In our setting, this is not the case - if an edge in the undirected graph is labelled $g \in G$, then the edge contributes to the final product as g itself *irrespective* of the direction that is taken by the path through the edge. Thus, the above results of the problem do not apply in our case.

2 Preliminaries

In this section, we list notations and preliminaries used in the paper. For standard notations and definitions of complexity classes, we refer the reader to the textbook [2]. We now define some graph theoretic terminologies required. A *tree decomposition* of a graph $G = (V, E)$ is a tree $T = (I, F)$ where each vertex $i \in I$ has a label $X_i \subseteq V$ with $\bigcup_{i \in I} X_i = V$ such that: for any edge $(u, v) \in E$, there exists an $i \in I$ with $u, v \in X_i$ and, for any $v \in V$, the vertices containing v in their label form a connected subtree of T . Given a tree decomposition $T = (I, F)$, the *width* of the decomposition is $\max_{i \in I} (|X_i|) - 1$. The *treewidth* of a graph G is the minimum k such that G has a tree decomposition of width k .

We define the algebraic structures that we refer to in the paper. Let \mathcal{A} be an algebraic structure where $*$ is the binary operator. If $*$ has the closure property, \mathcal{A} is said to be a groupoid. Groupoids for which $*$ is associative are called semi-groups. Semi-groups which have an identity element e (that is, $\forall a \in \mathcal{A}, ae = ea = a$)³ are called monoids. Groups are monoids for which every element has an inverse with respect to $*$. That is, $\forall a \in \mathcal{A}, \exists b \in \mathcal{A}$ such that $ab = ba = e$. In general, a monoid M is said to be divided by another monoid N if there exists a surjective morphism from a submonoid of M to N [22]. Quasigroups are a generalization of groups in a different direction; the operation in a quasigroup need not be associative but they are left and right cancellative (that is, $ab = ac \Rightarrow b = c$ and $ab = cb \Rightarrow a = c$).

\mathcal{A} -REACH Problem: Let $\mathbf{A} = \{(\mathcal{A}, *)\}$ be an infinite collection of algebraic structures where each $(\mathcal{A}, *)$ is the algebraic structure with set of elements $[k] = \{1, 2, \dots, k\}$ and the binary operation $*$ defined over \mathcal{A} . Let F be a subset of \mathcal{A} . Consider a graph $G = (V, E)$ and a function $\phi : E \rightarrow \mathcal{A}$. We extend the definition of ϕ to the yield of a path $p = v_0, v_1, \dots, v_m$, as $\phi(p) = \prod_{i=0}^{m-1} \phi((v_i, v_{i+1}))$ where product is the operation $*$ on the concatenated labels of p .

Definition 1. (\mathcal{A} -REACH) *Fix an algebraic structure \mathcal{A} . The \mathcal{A} -REACH problem asks: given a graph G on n vertices and the labelling function ϕ for the edges, two nodes s, t and accepting set⁴ $F \subseteq \mathcal{A}$, test whether there is a path p (need not be simple) from s to t such that $\phi(p) \in F$.*

³ We do not use the operator, whenever it is clear from the context. We use 1 and e interchangeably for the identity element.

⁴ If the size of \mathcal{A} is fixed (or even polynomially bounded) we will assume that $|F| = 1$. We also assume that the accepting element a is given as a part of the input. All our results except Theorem 7 hold even if a is fixed apriori.

For studying the variants of the problem, we introduce the following notation: A-GREACH refers to the \mathcal{A} -REACH problem defined over the algebraic structure A (Monoid, Aperiodic Monoid, Commutative Aperiodic Monoid, Group, Quasigroup and Semigroup) and the input graphs are restricted to the class G (Tree, Planar, DAG, k -Treewidth, Undirected(U) and Bidirected(B)).

Aperiodic Monoids and Quasigroups: The monoid class \mathbf{DA} is defined as the class of monoids that satisfy $(stu)^nt(stu)^n = (stu)^n$, for some n , for all s, t, u in the monoid. A language $L = A_0^*a_1A_1^*a_2 \cdots a_kA_k^*$ is said to be unambiguous if for all $w \in L$, there is a unique factorization $w = w_0a_1w_1a_2 \cdots a_kw_k$, such that $w_i \in A_i^*$ for $i = 0, 1, \dots, k$. Pin *et al.* [16] showed the following characterization:

Proposition 1. [16] *$L \subseteq A^*$ is recognized by a monoid in \mathbf{DA} if and only if L is a disjoint, finite union of unambiguous products $A_0^*a_1A_1^* \cdots a_kA_k^*$, where $A_i \subseteq A, a_i \in A$, for $i \in [k]$.*

The following was proved by Raymond *et al.* [17, 22].

Proposition 2. [17, 22] *Let M be a finite, non-commutative monoid. Then M is divided by one of the following aperiodic monoids. (1) BA_2 , the syntactic monoid⁵ of $(c^*ac^*bc^*)^*$. (2) U , the syntactic monoid of $((b+c)^*a(b+c)^*b(b+c)^*)^*$. (3) The syntactic monoid of $A^*aA^*bA^*$. (4) The syntactic monoid of c^*aA^* or A^*ac^* . Moreover, if $M \notin \mathbf{DA}$, M is divided by either BA_2 or U .*

In [4], Beaudry *et al.* define a *comb* as a left to right bracketing over a word, and claim that any bracketing over the word, for a quasigroup, can be viewed as a finite tree with each leaf as a comb. We state this as the following:

Proposition 3. [4] *Let $q_1q_2 \cdots q_n$ be a word over a quasigroup Q . If there is a bracketing such that $q_1q_2 \cdots q_n$ evaluates to q under that bracketing, then there is a bracketing with at most $8^{|Q|}$ combs which yields q .*

3 Logspace Upper Bounds

In this section, we explore the algebraic structure of the label set and graphs which enable us to solve the problem in L . As our main tool, we introduce the product graph, which is inspired by that of product graphs defined in the context of L-REACH problem by Yannakakis [23].

Product Graph and Properties: Let $G = (V, E)$ be a labelled graph, with a labelling $\phi : E \rightarrow M$, where M is a semigroup. We construct a new directed graph $G' = (V', E')$ as follows. We set $V' = V \times M$, and define the edge set E' as $\{((v_1, m_1), (v_2, m_2)) \mid (v_1, v_2) \in E \text{ and } m_1\phi(v_1, v_2) = m_2\}$. We show the following proposition. The proof is given in the full version.

Proposition 4. *For $s, t \in V, m \in M$, there is a path p from s to t in G such that $\phi(p) = m \iff$ there is a path from (s, e) to (t, m) in G' .*

⁵ We denote the elements of this monoid by $\{1, \alpha, \beta, \alpha\beta, \beta\alpha, 0\}$.

Similarly, we can argue that a path from (s, m_1) to (t, m_2) in G' exists if and only if there is a path from s to t in G , yielding m such that $m_1 m = m_2$.

NL Upper Bounds: Since the above proposition holds even if G has cycles in it, this implies that SEMIGROUPREACH is in NL. In later sections, we show more properties of the product graph. The product graph of an undirected graph labelled with a group is Eulerian. See Theorem 2. Also, the product graph of a graph with a bounded treewidth (labelled with a finite monoid) also has bounded treewidth. See Theorem 1.

If the algebraic structure is non-associative, we have to deal with all possible bracketings. Let us denote the set of all elements obtained by different bracketings of a word w by $\text{Yield}(w)$. Caussinus and Lemieux [8] showed that languages recognized by finite quasigroups are regular. Hence, there exists a morphism ψ from any quasigroup Q to a monoid M , and subsets $Q' \subseteq Q, M' \subseteq M$ such that for any word $w \in Q^*$, $\text{Yield}(w) \cap Q' \neq \emptyset$ if and only if $\psi(w) \in M'$. Hence, the product graph construction shows that QUASIGROUPREACH can be solved in NL.

Original Graph vs Product Graph While Group Labelling: It is a natural question to ask when the original graph appears as a subgraph in the product graph. We answer this for group labelled graphs.

Let $G = (V, E)$ be a directed acyclic graph, labelled with a group H , via labelling ϕ . Let the product graph be G' . Suppose G is a tree. We show that G' contains a copy of G . Let us start with any vertex v . For any $g \in H$, (v, g) is in G' . Now, consider all neighbors of v in G . If (v, u) is an edge in G , we have a corresponding edge $((v, g), (u, g\phi(v, u)))$ in G' . Similarly, if (u, v) is an edge in G , $((u, g\phi(u, v)^{-1}), (v, g))$ is an edge in G' . Continuing in a breadth first search manner, we get a copy of G in G' . Hence, it is easy to see that if G is a tree, G' contains G as a subgraph. To extend this to general DAGs, we need to understand when (undirected) cycles of G appear in G' . If all cycles (undirected) of G appear in the product graph, G also appears in the product graph.

Let $C = v_1, v_2, \dots, v_k$ be an undirected cycle in G . We define $\psi(v_i, v_j)$ as follows: $\psi(v_i, v_j) = \phi(v_i, v_j)$ if $(v_i, v_j) \in E$ and $\phi(v_j, v_i)^{-1}$ if $(v_j, v_i) \in E$. A proof of the following proposition is given in the full version.

Proposition 5. G appears as a subgraph in G' if and only if for each cycle $C = v_1, v_2, \dots, v_k$ in G , $\prod_{i=1}^k \psi(v_i, v_{i+1}) = \psi(v_k, v_1)^{-1}$.

Bounded Treewidth and Monoid Labelling: Das *et al.* [10] showed that reachability in bounded treewidth graphs can be tested in L. We show that, when a bounded treewidth graph G is labelled with a constant sized monoid M , the product graph of G still has constant treewidth, and hence, reachability in the labelled graph is also in L. The full version contains a detailed proof.

Theorem 1. MONOID- k -TREEWIDTHREACH is in L.

Group Labelled Graphs: Now we show that the \mathcal{A} -REACH problem can be solved in L, when the graph is undirected, and labelled with elements of a group, when the group size is polynomial in the size of the graph.

Theorem 2. GROUP-UREACH is L-complete.

Proof. To show that GROUP-UREACH is in L, we reduce the problem to REACH on Eulerian graphs, by showing that the product graph G' is Eulerian. From [19] we know that this problem can be solved in L and hence, this is sufficient. To solve this in L, Reingold *et al.* [19] observed (without proof) that, when each component of the given directed graph is Eulerian, a directed edge can be replaced by an undirected edge, and this does not alter connectivity of the graph. A proof can be found in the full version of this paper.

To show that G' is Eulerian, consider an edge (v_i, v_j) in G . Let $\phi((v_i, v_j)) = g$. Each vertex (v_i, g_k) , is hence connected to $(v_j, g_k g)$. We notice that for each k , $g_k g$ defines a different element in H . Similarly, each vertex (v_j, g_ℓ) is adjacent to $(v_i, g_\ell g)$. Hence, the edge (v_i, v_j) in G corresponds to $2|H|$ edges in G' , and these edges are such that each vertex of the form (v_i, g_k) and (v_j, g_ℓ) each have an indegree of 1 and an outdegree of 1. Since each edge in G increases the indegree and outdegree of any vertex in G' by the same amount, G' is Eulerian. Using the result from [19] we see that GROUP-UREACH is in L. To show hardness, we see that GROUP-UREACH is the undirected reachability problem when the underlying group is trivial. Hence, GROUP-UREACH is complete for L. \square

Observing that, for any $g \in G$, $g_k g$, is a different element for all k , and that each edge (v_i, v_j) in G gives rise to one incoming and one outgoing edge for each (v_i, g_k) holds even when the graph is bidirected. Hence, we conclude the following corollary.

Corollary 1. GROUP-BREACH is L-complete.

Logspace Algorithm for QUASIGROUP-UREACH: We notice from the proof of Theorem 2, that the product graph G' is Eulerian if the H has right cancellation, that is, if $ab = cb \Rightarrow a = c$. Since this holds for quasigroups as well, the constructed graph is Eulerian when H is a quasigroup. However, since evaluation of a word is over all possible bracketings, checking for a path from (s, e) to (t, h) is no longer sufficient (since this would correspond to only checking a left to right bracketing). Proposition 3 is used to prove the following theorem. The full version of this paper contains the proof.

Theorem 3. QUASIGROUP-UREACH is L-complete.

4 Symmetrizing by Labelling

In this section, we explore the question of whether we can reduce (in logspace) reachability over directed acyclic graphs to labelled reachability over undirected paths. We call this task as *symmetrization by labelling*. We first observe that symmetrization can be done when the algebraic structure is a specific aperiodic monoid or a specific, finitely generated, matrix group over \mathbb{Q} .

Labelling with Aperiodic Monoids: We give a labelling with a non-commutative aperiodic monoid, which makes the \mathcal{A} -REACH problem NL-hard.

In [15], Komarath *et al.* give a labelling with $(ab)^*$, for all directed acyclic graphs. We show that the syntactic monoid of this language is aperiodic. To see that it is aperiodic, we verify that for all a in the monoid, $a^3 = a^2$. Hence, this monoid is aperiodic with index 2. We also observe that the monoid is non-commutative.

Labelling with a Finitely Presented Group: In this subsection, we show that for matrix groups (even of size 2) with entries from \mathbb{Q} , symmetrization can be done. In Sect. 3, we saw that if symmetrization is done when the algebraic structure is either a polynomially growing group or a fixed size quasigroup, it implies that $NL = L$.

Theorem 4. *\mathcal{A} is the group of invertible $k \times k$ matrices with rational entries. \mathcal{A} -BREACH is NL-hard.*

Proof. We first show this for $k = 2$. We work over the following subgroup, $H = \left\{ \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} : \alpha \in \mathbb{Z} \right\}$. This group is finitely generated, since $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$ generate H . We define element $a = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$. Given an instance $(G(V, E), s, t)$ of REACH we construct an instance $(G'(V', E'), H, s, t, e)$ of GROUP-BREACH as follows. For every edge $(v_i, v_j) \in E$, we add 2 edges (v_i, v_j) and (v_j, v_i) to E' . We label edge (v_i, v_j) with e , and edge (v_j, v_i) with a .

We now argue correctness of this construction. Suppose there was a directed path from s to t in G . Let this path be $v_0 = s, v_1, v_2, \dots, v_k = t$. Now, in G' , we have the same path. Moreover, since each edge within the path is labelled with e , the entire path multiplies out to the identity element. Thus, we have a path from s to t in G' whose yield is identity.

Suppose there is no path from s to t in G , but there is a path from s to t in G' which yields identity. Let this path be $s = v_0, v_1, v_2, \dots, v_m = t$. Since this path does not exist in G , there must be an i such that $(v_i, v_{i+1}) \notin E$. Hence, the label on this edge must be a . We can have several edges like this in the path. Thus, the yield of the path is a^k , for some $k \geq 1$. Since the path yields identity, we have

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

However, we see that $a^k = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix}$. Hence, the yield cannot be identity, and there is no path from s to t in G' yielding identity.

To extend this to $k \times k$ matrices, we notice that we can embed a into a $k \times k$ matrix b by setting

$$b[i, j] = \begin{cases} a[i, j], & \text{if } i \leq 2, j \leq 2 \\ 1, & \text{if } i, j > 2, \text{ and } i = j \\ 0, & \text{if } i, j > 2, \text{ and } i \neq j \end{cases}$$

The forward direction of the proof is easy to see. The reverse direction follows from the fact that b^k can never be identity, for $k > 1$. □

5 A Dichotomy Theorem for Aperiodic Labelling

In this section, we prove the main result of the paper, which is the dichotomy theorem for finite aperiodic monoids with respect to the reachability in labelled graphs. We first settle the complexity in the case of commutative monoids. For non-commutative monoids, we show the dichotomy using the classification of aperiodic monoids by [17, 22] (see Proposition 2). We show deterministic logspace algorithms when the monoid is in DA and for the other cases (when the monoids are divided by U or BA_2), we show that the reachability is complete for NL.

Logspace Algorithm for COMMUTATIVEAPERIODIC-UREACH:

Let $M = \{1, \alpha_1, \alpha_2, \dots, \alpha_k\}$ be a commutative aperiodic monoid.

Theorem 5. *Let G be an undirected graph labelled with elements from M , a commutative aperiodic monoid. Checking if there is a path from s to t which evaluates to an element α can be done in L.*

Proof. We notice that any element α in M can be thought of as several tuples of integers (n_1, n_2, \dots, n_k) , such that $\alpha = \alpha_1^{n_1} \alpha_2^{n_2} \dots \alpha_k^{n_k}$. Hence, checking if a path evaluates to particular element is equivalent to checking if the the number of occurrences of each element in each path is one of the tuples associated with the element. We also know that M is aperiodic with index q ($\forall \alpha \in M, \alpha^{q+1} = \alpha^q$). This implies that, if $\alpha = \alpha_1^{n_1} \alpha_2^{n_2} \dots \alpha_i^q \dots \alpha_k^{n_k}$, then $\alpha = \alpha_1^{n_1} \alpha_2^{n_2} \dots \alpha_i^{q+1} \dots \alpha_k^{n_k}$. Hence, checking for tuples where each value is bounded by q is sufficient.

Let $G = (V, E)$ be the given graph, labelled with a commutative aperiodic monoid M , via a mapping ϕ . Let $s, t \in V$ and (n_1, n_2, \dots, n_k) be a tuple, where $n_i \leq q, \forall i$. Let $N = \sum_i n_i$. The algorithm does the following.

Repeat the following for each $(u_1, v_1), (u_2, v_2), \dots, (u_N, v_N) \in E^N$, such that the labels of $(u_1, v_1), (u_2, v_2), \dots, (u_N, v_N)$ form the tuple (n_1, n_2, \dots, n_k) . Set $v_0 = s, u_{N+1} = t$. Let $P = \{e\} \cup \{\alpha_i \mid n_i = q\}$. Let G' be G without edges labelled with any element from $M \setminus P$. We accept if there is a path from v_i to u_{i+1} , in G' for all i .

We see that the algorithm uses only logspace, since N is at most qk .

Correctness: The algorithm iterates over all possible edges, such that the labels of the edges give the tuple. For each set of edges, it verifies if there is a path between these edges, which uses only those labels which have crossed the index (captured by set P). This ensures that the resulting path also evaluates to the same element.

For the reverse direction, since every graph accepted by this algorithm has a path whose tuple is of the form $(n'_1, n'_2, \dots, n'_k)$, where $n'_i = n_i$ if $n_i < q$, and $n'_i \geq n_i$ if $n_i = q$. Hence, the elements that both these tuples evaluate to must be the same. □

Logspace Algorithm for DA-UREACH: We give a logspace algorithm to solve DA-UREACH, when the graph is labelled with letters from an unambiguous concatenation $L = A_0^* a_1 A_1^* a_2 \dots a_k A_k^*$, where A is the alphabet, $A_i \subseteq A, a_i \in A, \forall i$. From Proposition 1, this is sufficient to show that DA-UREACH can be solved in logspace.

Theorem 6. *Let G be an undirected graph labelled with elements from an alphabet A . Let s and t be given vertices in G . Let $L = A_0^*a_1A_1^*a_2A_2^*\cdots a_kA_k^*$ be an unambiguous concatenation, where $A_i \subseteq A, a_i \in A, \forall i$. Checking if there is a path from s to t , whose yield is in L can be done in logspace.*

Proof. Let $G = (V, E)$ be the given graph, labelled with an alphabet A , via a mapping ϕ . Let $L = A_0^*a_1A_1^*a_2\cdots a_kA_k^*$. Let $s, t \in V$. The algorithm does the following: Repeat the following for $(u_1, v_1), (u_2, v_2), \dots, (u_k, v_k) \in E^k$, such that $\forall i, \phi(u_i, v_i) = a_i$. Set $v_0 = s, u_{k+1} = t$. For each i , let G_i be G without edges labelled with any element from $A \setminus A_i$. We accept if there is a path from v_i to u_{i+1} , in G_i for all i . Since k is finite, we see that the algorithm chooses all possible edges for the a_i 's, and check if paths between these edges are in A_i^* . The algorithm uses only logspace. The correctness of this algorithm is easy to see - if there exists a path from s to t in L , the algorithm will eventually find it, since it runs over all possible edges. For the other direction, we notice that the algorithm only accepts paths in L . □

Labelling with Non-commutative Aperiodic Monoids not in DA: We show that labelling an undirected graph with either BA_2 or U makes the \mathcal{A} -REACH problem NL-hard. Komarath *et al.* [15] give a labelling with $(ab)^*$, for all directed acyclic graphs. This immediately gives us a labelling with BA_2 . We give a similar labelling with U . We know that any non-commutative aperiodic monoid M not in **DA** is divisible by either U or BA_2 . Hence, we have a surjective morphism from a submonoid of M to either U or BA_2 . We show that labelling an undirected graph with BA_2 or U makes the \mathcal{A} -REACH problem NL-hard. By using the morphism, we can get instances of \mathcal{A} -REACH problem over undirected graphs, labelled with M , which are NL-hard.

Theorem 7. *\mathcal{A} -REACH for undirected graphs is NL-complete when the graph is labelled with U .*

Proof. We give a labelling from $L = (b^*ab^*bb^*)^*$ (whose syntactic monoid is U) similar to that in [15]. Let $G = (V, E)$ be a directed acyclic graph, with vertices s and t . Without loss of generality, we assume that s is a source (that is, it has only outgoing edges). We create a labelled, undirected graph $G' = (V', E')$ as follows. Each vertex in V is copied to V' . Additionally, for each directed edge (v_i, v_j) , we add a vertex m_{ij} to V' . Edges and labels are constructed as follows. If (v_i, v_j) is an edge in G , $(v_i, m_{ij}), (m_{ij}, v_j)$ are edges in G' , with (v_i, m_{ij}) being labelled with b , and (m_{ij}, v_j) is labelled with a . That is, we split each edge, labelling the first half with b , and the second half with a . We also add a new vertex t' , and add an edge (t, t') , labelled with b . We claim that there is a path from s to t in G if and only if there is a path from s to t' in G' , whose yield is in L .

The forward direction is easy to see. Suppose there is a path from s to t in G . Let the path be $s = v_{i_1}, v_{i_2}, \dots, v_{i_m} = t$. We claim that the path $s = v_{i_1}, m_{i_1i_2}, v_{i_2}, m_{i_2i_3}, v_{i_3}, \dots, m_{i_{m-1}i_m}, v_{i_m} = t, t'$ exists in G' and the yield of the path is in L . By our construction, each of these edges exist in G' . To see the yield, we notice that since $(v_{i_\ell}, v_{i_{\ell+1}})$ is in E , $(v_{i_\ell}, m_{i_\ell i_{\ell+1}})$ is labelled with b ,

whereas $(m_{i_\ell i_{\ell+1}}, v_{i_{\ell+1}})$ is labelled with a , for all ℓ . Hence, the yield is $(ba)^m b$ which is in L .

Suppose we do not have a path from s to t in G , but there is a path from s to t' in G' with a yield in L . Since there is no path in G , the path in G' must have taken some edges incorrectly. Let (u, v) be the first incorrect edge taken. That is, suppose $(v_i, v_j) \in E$. The edge taken is either of the form (v_j, m_{ij}) or (m_{ij}, v_i) . For the first, the yield up to this point is $(ba)^\ell$, for some ℓ , and the edge is labelled with a . This results in two consecutive a 's, which cannot be in the language, and the path in G' cannot have any edge of this form. For the second, we see that since (m_{ij}, a_i) is the first incorrect edge taken, the edge taken before this is (a_i, m_{ij}) , and both these edges can be ignored. Thus, if all incorrect edges taken are of the second form, we can create a path from s to t in G , contradicting our initial assumption. \square

6 Planarizing by Labelling

We now present a reduction from the reachability problem to \mathcal{A} -REACH over planar DAGs when \mathcal{A} is the fixed monoid BA_2 . The same reduction can be achieved with group labelling when the size of the groups is allowed to be exponentially growing. We give a reduction from REACH to MONOID-PLANARREACH and hence it is NL-hard.

Theorem 8. *Let $G = (V, E)$ be a graph. Let $\phi : E \rightarrow BA_2$ be a labelling function. Then REACH reduces to MONOID-PLANARREACH.*

A proof of this theorem is present in the full version. We make an observation that the hard instances to MONOID-PLANARREACH have the property that the underlying undirected graph can be bipartite. In a close contrast to the results in the previous section, we show that if the labels are coming from BA_2 , and in particular from the set $\{\alpha, \beta\}$ and the graph is bipartite, then $NL = UL$. That is, if the labelling had preserved bipartiteness of the graph (which we can ensure in the reachability instances by subdividing every edge into two edges by introducing an intermediate vertex), then $NL = UL$. The proof of the following theorem is present in the full version.

Theorem 9. *Let $G = (V, E)$ be a planar graph whose underlying undirected graph is bipartite, and labelled with BA_2 with $\phi : E \rightarrow \{\alpha, \beta\}$. The \mathcal{A} -REACH problem (between any two vertices) in G can be reduced (in log-space) to testing reachability in planar DAGs and hence is in UL.*

Following the quest for more structure in the labelling set, we now give a reduction from REACH to GROUP-PLANARREACH, when labelled with a group having size exponential in the size of the graph, thus showing that it is NL-hard. A proof for the following theorem is present in the full version.

Theorem 10. *GROUP-PLANARREACH is NL-hard when the group size is $\Omega(2^{n^4})$ where n is the size of the graph.*

References

1. Allender, E.: Reachability problems: an update. In: Cooper, S.B., Löwe, B., Sorbi, A. (eds.) CiE 2007. LNCS, vol. 4497, pp. 25–27. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-73001-9_3](https://doi.org/10.1007/978-3-540-73001-9_3)
2. Arora, S., Barak, B.: Computational Complexity: A Modern Approach. Cambridge University Press, Cambridge (2009)
3. David, A., Barrington, M., Thérien, D.: Finite monoids and the fine structure of NC^1 . J. ACM **35**(4), 941–952 (1988)
4. Beaudry, M., Lemieux, F., Thérien, D.: Finite loops recognize exactly the regular open languages. In: Degano, P., Gorrieri, R., Marchetti-Spaccamela, A. (eds.) ICALP 1997. LNCS, vol. 1256, pp. 110–120. Springer, Heidelberg (1997). doi:[10.1007/3-540-63165-8_169](https://doi.org/10.1007/3-540-63165-8_169)
5. Beaudry, M., McKenzie, P., Pladeau, P., Thérien, D.: Finite monoids: from word to circuit evaluation. SIAM J. Comput. **26**(1), 138–152 (1997)
6. Bédard, F., Lemieux, F., McKenzie, P.: Extensions to Barrington’s M-program model. In: Proceedings Fifth Annual Structure in Complexity Theory Conference, pp. 200–209 (1990)
7. Bourke, C., Tewari, R., Vinodchandran, N.V.: Directed planar reachability is in unambiguous log-space. ACM Trans. Comput. Theory **1**(1), 4 (2009)
8. Caussinus, H., Lemieux, F.: The complexity of computing over quasigroups. In: Thiagarajan, P.S. (ed.) FSTTCS 1994. LNCS, vol. 880, pp. 36–47. Springer, Heidelberg (1994). doi:[10.1007/3-540-58715-2_112](https://doi.org/10.1007/3-540-58715-2_112)
9. Chandra, A.K., Fortune, S., Lipton, R.J.: Unbounded fan-in circuits and associative functions. J. Comput. Syst. Sci. **30**(2), 222–234 (1985)
10. Das, B., Datta, S., Nimbhorkar, P.: Log-space algorithms for paths and matchings in k-trees. In: 27th STACS, pp. 215–226 (2010)
11. Geelen, J., Gerards, B.: Excluding a group-labelled graph. J. Comb. Theory Ser. B **99**(1), 247–253 (2009)
12. Horwitz, S., Reps, T.W., Binkley, D.: Interprocedural slicing using dependence graphs. ACM Trans. Program. Lang. Syst. **12**(1), 26–60 (1990)
13. Huynh, T.C.T.: The linkage problem for group-labelled graphs. Ph.D. thesis, University of Waterloo (2009)
14. Kawase, Y., Kobayashi, Y., Yamaguchi, Y.: Finding a path in group-labeled graphs with two labels forbidden. In: Halldórsson, M.M., Iwama, K., Kobayashi, N., Speckmann, B. (eds.) ICALP 2015. LNCS, vol. 9134, pp. 797–809. Springer, Heidelberg (2015). doi:[10.1007/978-3-662-47672-7_65](https://doi.org/10.1007/978-3-662-47672-7_65)
15. Komarath, B., Sarma, J., Sunil, K.S.: On the complexity of L-reachability. Fundam. Inform. **145**(4), 471–483 (2016)
16. Pin, J.-E., Straubing, H., Thérien, D.: Locally trivial categories and unambiguous concatenation. J. Pure Appl. Algebra **52**(3), 297–311 (1988)
17. Raymond, J.-F.Ç., Tesson, P., Thérien, D.: An algebraic approach to communication complexity. In: Larsen, K.G., Skyum, S., Winskel, G. (eds.) ICALP 1998. LNCS, vol. 1443, pp. 29–40. Springer, Heidelberg (1998). doi:[10.1007/BFb0055038](https://doi.org/10.1007/BFb0055038)
18. Reingold, O.: Undirected connectivity in log-space. J. ACM **55**(4), 17 (2008)
19. Reingold, O., Trevisan, L., Vadhan, S.: Pseudorandom walks on regular digraphs and the RL vs. L problem. In: Proceedings of STOC, pp. 457–466 (2006)
20. Reps, T.W.: On the sequential nature of interprocedural program-analysis problems. Acta Inform. **33**(8), 739–757 (1996)

21. Reps, T.W.: Program analysis via graph reachability. *Inf. Softw. Technol.* **40**(11–12), 701–726 (1998)
22. Tesson, P.: An algebraic approach to communication complexity. Masters thesis, McGill University, Montreal (1998)
23. Yannakakis, M.: Graph-theoretic methods in database theory. In: *Proceedings of the 9th ACM PODS*, pp. 230–242 (1990)