UPPER BOUNDS FOR MONOTONE PLANAR CIRCUIT VALUE AND VARIANTS

Nutan Limaye, Meena Mahajan, and Jayalal Sarma M.N.

Abstract. The P-complete Circuit Value Problem CVP, when restricted to monotone planar circuits MPCVP, is known to be in NC^3, and for the special case of upward stratified circuits, it is known to be in $\mathsf{LogDCFL}$. In this paper we re-examine the complexity of MPCVP, with special attention to circuits with cylindrical embeddings. We characterize cylindricality, which is stronger than planarity but strictly generalizes upward planarity, and make the characterization partially constructive. We use this construction, and four key reduction lemmas, to obtain several improvements. We show that stratified cylindrical monotone circuits can be evaluated in $\mathsf{LogDCFL}$, and arbitrary cylindrical monotone circuits can be evaluated in $\mathsf{AC}^1(\mathsf{LogDCFL})$, while monotone circuits with one-input-face planar embeddings can be evaluated in LogCFL. For monotone circuits with focused embeddings, we show an upper bound of $\mathsf{AC}^1(\mathsf{LogDCFL})$. We re-examine the NC^3 algorithm for general MPCVP, and note that it is in $\mathsf{AC}^1(\mathsf{LogCFL}) = \mathsf{SAC}^2$. Finally, we consider extensions beyond MPCVP. We show that monotone circuits with toroidal embeddings can, given such an embedding, be evaluated in NC. Also, special kinds of arbitrary genus circuits can also be evaluated in NC. We also show that planar non-monotone circuits with polylogarithmic negation-height can be evaluated in NC.

Keywords. circuit value, monotone, planar, NC

Subject classification. Computational and Structural Complexity

1. Introduction

Given a Boolean circuit C over n inputs $x_1, \ldots, x_n$, and an assignment $x_i = a_i$ for each variable x_i, the Circuit Value Problem CVP is to determine the value $C(a_1, \ldots, a_n)$. This is a fundamental problem in complexity theory, since circuits capture computation in a very natural and universal way. When each gate is labeled AND, OR or NOT, CVP is complete for the complexity class P. It remains complete if the circuits are monotone (no NOT gates); it also remains complete if the underlying graph has a planar embedding. However, if the circuit is simultaneously monotone and planar (MPCVP), then evaluating it is in NC.

The history of MPCVP begins with the papers of Goldschlager, where it is shown that planar CVP and monotone CVP are P-complete [Gol77], and that a special case of MPCVP, upward stratified (see Section 2 for a formal definition) is in NC^2 [Gol80]. Subsequently, Dymond and Cook [DC89] improved the upper bound for this special case to LogCFL, and Kosaraju [Kos90] extended the result by showing that a less restrictive special case, namely that of layered upward planar monotone circuits (subsuming Goldschlager's case), is also in NC, in fact in NC^3. Independently and in parallel, Delcher and Kosaraju [DK95] and Yang [Yan91] showed that MPCVP in its full generality is in NC^4 and in NC^3 respectively. More recently, Barrington, Lu, Milterson and Skyum [BLMS99] showed that for monotone upward stratified circuits — the special case considered in [Gol80, DC89] — there is in fact an upper bound of LogDCFL. Here, L and NL stand for deterministic and non-deterministic logspace respectively, while LogDCFL and LogCFL stand for the classes of languages logspace-many-one-reducible to deterministic and arbitrary context-free languages respectively. Recall that $\mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{LogCFL}$, $\mathsf{L} \subseteq \mathsf{LogDCFL} \subseteq \mathsf{LogCFL}$, and $\mathsf{LogCFL} = \mathsf{SAC}^1 \subseteq \mathsf{AC}^1 \subseteq \mathsf{NC}^2$. See any standard text on circuit complexity (e.g. [Vol99]) for more details.

There has recently been a spurt of activity examining topological constraints in small-width circuits [BLMS99, HMV06, Han06, ADR05b]. These works provide more insights into how to exploit the restricted topology. Using these insights, we review the developments leading up to and beyond the "MPCVP is in NC" result, and make some improvements on the known bounds for general MPCVP as well as some special cases. (However, we do not consider width restrictions in this work.) Our main improvements are obtained while considering circuits with cylindrical embeddings. Such embeddings strictly subsume upward planar embeddings, but are not strong enough to capture all of planarity. They have been studied in depth in the context of small-width circuits in [HMV06, Han06]. Another major contribution we obtain is to extend the NC upper bound on MPCVP to toroidal (genus one) monotone circuits.

A key limiting problem that arises in our constructions is that of finding the length of a longest path in a planar directed acyclic graph (planar DAG). We define PDLP to be the class of problems logspace many-one reducible to this problem. While finding longest paths in general is hard, finding longest paths in DAGs is easily seen to be in NL, and in fact, NL-complete. It is conceivable, however, that the longest path problem over planar DAGs is considerably easier than NL. Hence when we need longest paths in planar DAGs, we state our upper bounds explicitly in terms of PDLP rather than NL, keeping in mind that $\mathsf{L} \subseteq \mathsf{PDLP} \subseteq \mathsf{L}(\mathsf{PDLP}) \subseteq \mathsf{L}(\mathsf{NL}) = \mathsf{NL}$.

Our contributions are as follows:

1. We characterize cylindrical graphs as spanning subgraphs of single-source single-sink planar DAGs (Theorem 3.4). This is implicit in the result of Hansen (Theorem 2 of [Han06]), where layered cylindrical graphs are characterized as subgraphs of single-source single-sink layered planar DAGs. We state it explicitly because we obtain a partial logspace-constructive version, even when the given DAG is not layered to begin with. (Layering, in general, could be harder than logspace.) These results are presented in Section 3.

2. We present four reduction lemmas (Lemma 4.1, 4.2, 4.3 and 6.2) which are at the heart of the improvements we obtain. The topological constraints considered are shown in Figure 1.1. The thick arrows go from stronger to weaker constraints, the dotted arrows indicate logspace reductions, and the dashed arrows indicate the reductions in L(PDLP).

3. Using the reduction lemmas, we obtain improved upper bounds; see Table 1.1.

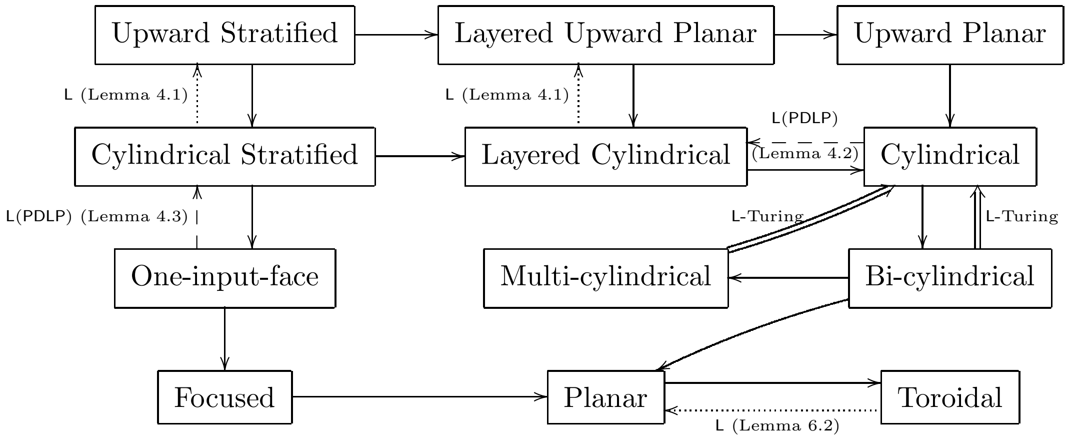


Figure 1.1: Relationship between various topological restrictions in the context of MCVP

4. We consider a restricted generalization to higher genus in Section 6.2 and show that such monotone circuits can be evaluated in NC.

5. We also consider planar non-monotone circuits with restrictions on the placement of negation gates, in Section 6.3, and show that such circuits too can be evaluated in NC.

(Monotone) Circuit type	Embedding	Our upper bound	Previous bound
Cylindrical stratified	given	$\mathsf{LogDCFL}$ (Thm. 5.1)	NC^2 ([Yan91] Sec. 2)
One input face	not needed	$\mathsf{L}(\mathsf{PDLP} \oplus \mathsf{LogDCFL})$ (Thm. 5.2)	NC^2 ([Yan91] Sec. 3)
Cylindrical	given	$\mathsf{AC}^1(\mathsf{LogDCFL})$ (Thm. 5.5)	–
Planar	not needed	$\mathsf{AC}^1(\mathsf{LogCFL}) = \mathsf{SAC}^2$ (Thm. 5.8)	NC^3 [Yan91]
Toroidal	given	$\mathsf{AC}^1(\mathsf{LogCFL}) = \mathsf{SAC}^2$ (Thm. 6.3)	P
Non-monotone planar, polylog negation-height	not needed	NC (Lem. 6.4,6.5)	P

Table 1.1: Improved upper bounds

2. Basic definitions

2.1. Circuits. The underlying graph of any circuit is a directed acyclic graph (DAG). We consider circuits with gates labeled AND, OR, NO-OP, 0, 1. A gate labeled AND or OR has fan-in two, a gate labeled NO-OP has fan-in one, and a gate labeled by a constant has fan-in zero and is a source node. Without loss of generality, we assume that constant gates have fan-out one and that no gate has fan-out greater than two. We do not assume that there is a single sink. The earlier NC algorithms for MPCVP made this assumption, since if there are multiple sinks, each of them can be evaluated independently. However, removing nodes with no path to the designated sink may not be possible in logspace, so we explicitly note this as a computational requirement.

A circuit with variables is a circuit in which some fan-in zero gates are labeled by variables. By generalized circuits we mean circuits which also have constant gates with non-zero fan-in and possible fan-out more than one; the output of such a gate is independent of its inputs, but the input wires could play a role in determining the planar embeddings. Generalized circuits, with or without variables, arise in the recursive steps of the algorithms from [DK95, Yan91].

A circuit is said to be *layered* if there is a partition $V = V_0 \cup V_1 \cup \ldots \cup V_h$ such that all edges go from some layer V_i to the next layer V_{i+1}. A circuit is said to be *stratified* if it is layered and all source nodes are in layer V_0.

A language L is said to be in NC if there is a family of polynomial-size polylog depth circuits $\{C_n\}$ with AND, OR, and NOT gates, with all NOT gates at the leaves, such that $x \in L$ iff $C_{|x|}(x) = 1$. Circuit C_n having depth $O(\log^i n)$ corresponds to NC^i if the AND/OR gates have bounded fan-in, to AC^i if they have unbounded fan-in, and to SAC^i if only the AND gates are constrained to bounded fan-in. Clearly, $\mathsf{NC}^i \subseteq \mathsf{SAC}^i \subseteq \mathsf{AC}^i \subseteq \mathsf{NC}^{i+1}$.

2.2. Topological Embeddings and Drawings. In this paper, we are concerned with directed acyclic graphs, denoted DAGs. Though many of the definitions below apply to general graphs, we will use them specialized to DAGs.

A graph is said to be *planar* if it can be embedded in the plane without crossings. That is, the nodes and edges of the graph can be drawn in such a way that the representations of no two edges intersect, except at shared endpoints. A plane graph is a graph along with a planar embedding. Note that planarity is independent of whether the graph is directed or not. By the results of [RR94, AM04, Rei05], deciding if a given graph is planar and if so finding a planar embedding is in AC^1, SL, and now L.

A planar embedding is *bimodal* if at every vertex v, all outgoing (incoming) edges appear consecutively around v. It is easy to see ([TT86], [Han06] Lemma 5, [Yan91] Lemmas 3.1 and 3.2) that in a planar DAG with a single source and a single sink, (a) every embedding is bimodal, and (b) for every face f, the edges incident on f form a simple (undirected) cycle consisting of two directed paths.

A planar embedding of a DAG is said to be a *one-input-face embedding* if all source nodes lie on the same face. Testing if a planar DAG is one-input-face, and if so, uncovering such an embedding, is easy: add a new source node with edges to all the old sources, and test for planarity.

A drawing (not necessarily planar) of a digraph on the plane is *upward* if the drawing of every edge is monotonically increasing in the vertical direction. Every DAG has an upward embedding, which can be recovered by a topological sort. (Also, only DAGs have upward embeddings, since a cycle cannot be embedded in an upward way.)

A digraph is *upward planar* if it has an embedding that is simultaneously upward and planar. Though all DAGs are upward, not all planar DAGs are upward planar. Figure 2.1 shows a standard instance of a planar DAG which is not upward planar (see for instance [BT88]). In fact, given a planar DAG, deciding whether it is upward planar is NP-complete [GT01]. (It is also known that every upward planar graph has an upward planar embedding using only straight-line drawings of all edges [BT88]. Furthermore, if

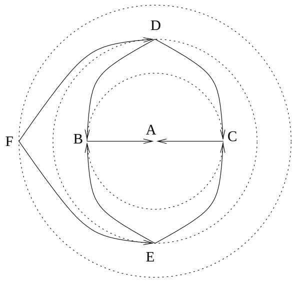


Figure 2.1: A planar DAG that is cylindrical but not upward planar

the DAG is layered, all nodes in the same layer will have the same y-coordinate.)

A digraph is *cylindrical* if it can be embedded on a cylinder surface, in a way such that all edges are monotonically increasing in the direction of the axis of the cylinder. (Clearly such a digraph must also be acyclic, a DAG.) As observed in [Han06], this generalizes upwardness, with the edges embedded on the surface of the cylinder rather than on a plane. Note that the surface of the cylinder can be embedded on a plane in a straightforward way: place the right end of the cylinder (the end towards which all edges flow) on the plane, and dilate the cylinder in a continuous way into a cone section until its surface lies flat around the end placed first. (In fact, the converse is also true: any embedding on the plane can be drawn on the surface of the cylinder. But the edges may not be monotone along the cylinder axis.) Thus a cylindrical embedding will give rise to a planar embedding where all edges flow in an inward direction towards a central face. It follows that every cylindrical embedding is also bimodal, even if it is not single-source single-sink.

Cylindricality strictly generalizes upward planarity, as Figure 2.1 shows. The example of Figure 2.2 shows that cylindricality does not capture all planar DAGs.

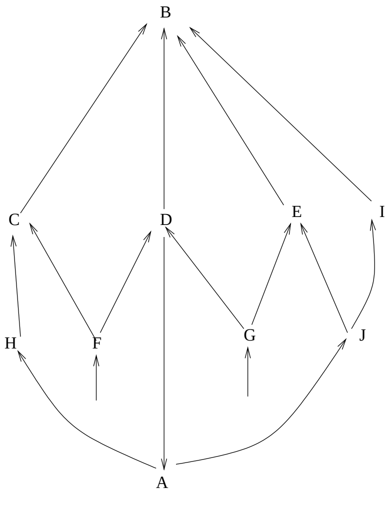


Figure 2.2: A planar DAG that is not cylindrical

A *layered cylindrical embedding* of a layered digraph is a cylindrical embedding where layers correspond to disjoint circles of the cylinder (or concentric circles on the plane, in the corresponding inward drawing). In recent literature in the graph drawing community, the term *radial drawing* is used. For instance, the radial leveled planar drawings of [BBF03] are exactly layered cylindrical embeddings. We continue to use the term cylindrical rather than radial, since the main issue in radial leveled planar drawings appears to be: given the partition of the vertex set into sets lying on the same layer, find the ordering on each layer. On the other hand, we are often concerned with finding the partition as well, and this could well be a harder problem.

Recall that a layered circuit (in general, a layered DAG) is said to be stratified if all source nodes appear at layer 0. A DAG is said to be *upward stratified* (*cylindrical stratified*) if it is layered, stratified, and has an upward planar (cylindrical respectively) embedding. It follows that an upward/cylindrical stratified circuit has a one-input-face embedding. Figure 2.3 shows a layered planar DAG which has an upward planar embedding and a one-input-face embedding but no upward one-input-face embedding. In [DK95], the term *restricted stratified* is used to denote circuits which are *cylindrical stratified* as defined above (without the *restricted*, the authors of [DK95] mean *generalized* circuits). On the other hand, in [BLMS99], *stratified* refers to *upward stratified* as described here.

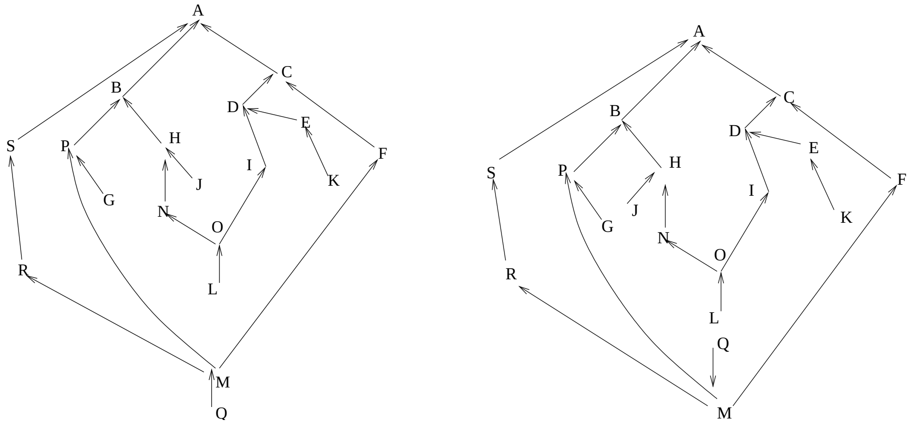


Figure 2.3: A layered planar DAG with an upward planar embedding and a one-input-face embedding but with no upward one-input-face embedding

A planar embedding of a DAG G is *focused* if there is a subset S of source nodes, all of which are embedded on a single face, and every node of G not reachable from S is itself a source node. This is a topological analogue of a skewness condition on circuits. Note that one-input-face embeddings are (vacuously) focused; S is the set of all source nodes.

We use the terms SSPD and SMPD to mean single-source single-sink planar DAGS and single-source multiple-sink or multiple-source single-sink planar DAGs respectively.

2.3. Representing embeddings.

Planar embeddings: By the results of [RR94, AM04, Rei05], deciding if a given graph is planar and if so finding a planar embedding is in AC^1, SL, and now L. The embedding so obtained is a planar combinatorial embedding, specifying the cyclic (clockwise, say) ordering of edges around each vertex in some plane embedding. (In fact, specifying for each vertex the clockwise cyclic ordering of edges around it is what is called a combinatorial embedding, and corresponds to an embedding of the graph on some orientable surface of appropriate genus.) Checking whether a given combinatorial embedding corresponds to an embedding on the plane can be done in logspace.

We briefly discuss how faces are specified in any planar embedding. Recall that embeddings ignore directions on edges. In fact, for each (undirected) edge (u, v), the embedding will specify where arc (u, v) figures in the circular list around u, and where arc (v, u) figures in the circular list around v. The arcs (u, v) and (v, u) are expected to be superimposed in the corresponding geometric embedding. We use the term edges to refer to directed edges of the original graph, while we use the term arcs to refer to the directed arcs in the combinatorial embedding. For every arc $e = (u, v)$, there are faces $L(e)$ and $R(e)$ to the left and right, respectively, of the edge. (These could both be the same, if, say, e is a bridge in the underlying graph.) If G is a connected graph when directions on edges are ignored, then for every face f, the set of edges e with $f \in \{L(e), R(e)\}$ form a connected graph. This set can be traversed systematically as follows. Start with an arc $e = (u, v)$ such that, say, $f = R(e)$. Let $e' = (v, w)$ be the arc preceding (v, u) in the cyclic ordering around v. Then $f = R(e')$. Keep advancing in this way until the starting arc is encountered again; in the process, the entire boundary of f will be traversed. We assume that f is "named" by the lexicographically smallest arc $a = (u, v)$ such that $f = R(a)$. See [MT01, Whi73] for more about representing embeddings.

Layered cylindrical or Layered upward planar embeddings: We assume that the embedding is given in the following form: (a) the cyclic ordering of edges around each vertex (the planar combinatorial embedding) corresponding to the geometric embedding, and (b) the circular or left-to-right ordering of vertices at each layer. It is straightforward to see that given such information, we can verify in logspace that it indeed corresponds to some layered cylindrical or layered upward planar geometric embedding.

Cylindrical embeddings: For cylindrical embeddings of non-layered graphs, we need to specify some more information. Imagine circles drawn along the surface of the cylinder, through each vertex. The ordering of the circles along the axis of the cylinder imposes a partial order on the vertices (total, if no two vertices lie on the same circle); consider any total order extending this. This ordering corresponds to non-decreasing distance of vertices from the bottom end of the cylinder. For each vertex u, we can talk of its left face and its right face: the left face is the face between u's leftmost incoming edge (last incoming edge in clockwise ordering) and leftmost outgoing edge (first outgoing edge in clockwise ordering), while the right face is the face between its rightmost incoming and outgoing edges. If u is a source, then the left and the right face are the same, and it is the face containing the (initial segment of) the ray drawn out of u against the cylinder axis. Similarly, if u is a sink, it is the face containing the (initial segment of) the ray drawn out of u along the cylinder axis. Given the clockwise ordering of edges around each vertex, the left and right faces can be determined for each u that is not a source or sink. For a source/sink u, if we explicitly specify the leftmost outgoing/incoming edge, then this face can be determined. We call this edge $L(u)$. For instance, see the example in Figure 2.4. The total order is $A\ B\ E\ F\ C\ D$. For source A, $L(A) = (A, B)$, while for sink D, $L(D) = (C, D)$. The left faces of B and C are f_l and f_r respectively. The right face of B is the region inside the quadrilateral $BFEA$, while the right face of C is the region inside the triangle BCD.

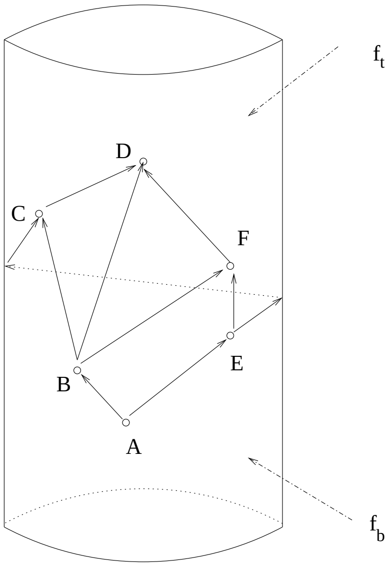


Figure 2.4: Representing a cylindrical embedding

With this background, we now assume that the following information about the cylindrical embedding is available: (a) the cyclic ordering of edges around each vertex (the planar combinatorial embedding), (b) a total order $v_1, v_2, \ldots, v_n$ of the vertices, extending the partial order induced by the cylindrical embedding, and (c) for each source/sink u, the edge $L(u)$. In particular, the edges $L(v_1)$ and $L(v_n)$ specify the faces f_b and f_t corresponding to the bottom and top ends of the cylinder.

It is not clear that given (a), (b), (c) above, one can check in logspace if the corresponding plane embedding is cylindrical. However, this information is sufficient for the results of this paper.

3. Graphs on cylinders

Upward planar graphs have been characterized independently in [Kel87] and [BT88]: A DAG is upward planar if and only if it is a subgraph of a planar st-digraph, that is, a planar DAG with a single source s, a single sink t, and an edge from s to t. Extending this result, [Han06] characterizes layered cylindricality: a layered digraph is layered cylindrical if and only if it is a subgraph of a layered planar DAG with a unique source and a unique sink (an SSPD). While the result is implicit in the work of [TT89], the major contribution in the proof of [Han06] is to make the transformation uniform. In a similar vein, we characterize cylindricality (without the layered property); while the topological ideas are already there in the proofs of [TT89, Han06], we prove it in a different way to obtain suitable uniformity bounds. We then use these to evaluate cylindrical circuits.

One direction of our characterization crucially uses a layered embedding algorithm independently due to [Yan91] and [DK95]. The algorithm of [Yan91] is stated for single-sink digraphs where there is a one-input-face planar embedding (an embedding in which all sources appear on the same face), while that of [DK95] is stated for what are called focused circuits. We will use the algorithm for single-sink one-input-face planar DAGs, and we observe that this includes, as a special case, SSPDs. ([Yan91] uses the notation layered one-input-face for cylindrical stratified (all source nodes at the first layer)). An important property of such embeddings is that all vertices are bimodal; thus left and right faces of a vertex are defined. The algorithm is described in Figure 3.1.

Input: a one-input-face single-sink planar directed acyclic graph H.

Output: A layered cylindrical embedding of a graph H', obtained from H by subdividing edges into directed paths.

Method: Let t be the sink of H.

1. For each node v in H find the longest distance $d(v)$ to t. Let $d = \max_v\{d(v)\}$; there are $d+1$ layers. The input nodes are in V_0. A non-input node u is in layer $l(u) = d - d(u)$.
2. For a directed edge (u, v) in the graph, let $k = l(v)-l(u)-1$. If $k > 0$, then introduce dummy nodes $n_1, n_2 \dots n_k$ and add the edges (u, n_1),$(n_1, n_2) \dots$ (n_k, v). (That is, we subdivide edge (u, v) into a directed path of length $l(v) - l(u)$.) The dummy node n_i will be in layer $l(u) + i$.
3. For each node u (including dummy nodes), walk along the boundary of the left (or right, respectively) face of u beginning at u. The first node encountered with the same layer number as u is the left (or right, respectively) neighbour of u.

Figure 3.1: Layered embedding algorithm ([Yan91] Section 3, [DK95] Section 4)

Steps 1-2 of the algorithm provide the layering, step 3 provides the cylindrical embedding of the layered graph. To see why the algorithm is correct, see Section 3 of [Yan91] or [DK95]. We observe the following:

Proposition 3.1. *The layered embedding algorithm above runs in* $\mathsf{L}(\mathsf{PDLP})$. □

Now we establish our characterization by the following two lemmas.

Lemma 3.2. *If a planar DAG G is a spanning subgraph of an SSPD H (a planar DAG with a single source and a single sink), then G has a cylindrical embedding which, given G and H, can be constructed in* $\mathsf{L}(\mathsf{PDLP})$.

Proof. Using the algorithm of Fig. 3.1, a cylindrical embedding can be found for H' obtained from H by edge subdivision. Replacing the directed paths obtained through subdivision by original edges, we get a cylindrical embedding of H, and hence of G. The upper bound for constructing the embedding of H follows from Proposition 3.1. □

Lemma 3.3. *If a planar DAG G has a cylindrical embedding, then it is a spanning subgraph of a cylindrical DAG H with a single source and a single sink.*

Proof. Consider the layout of the graph on the cylinder surface, with vertices in order $v_1, v_2, \ldots, v_n$ as specified by the cylindrical embedding. Clearly, v_1 is a source and v_n is a sink. Without loss of generality, we assume that the normal circles of the cylinder through v_1 and v_n do not contain any other vertex. (If they do, move vertex v_1 slightly towards the cylinder bottom, v_n towards the top. This does not change the combinatorial specification of the embedding.)

If any vertex v_i other than v_n is a sink, we need to add an edge from it to some v_j with $j \geq i$ without destroying cylindricality. Such a v_j can always be found as follows: imagine a particle moving out of v_i along the direction of the cylinder axis. It aims to avoid intersecting any edge. So if it encounters an edge, it moves parallel to and infinitesimally close to the edge. Since all edges are cylindrical, its movement is still monotonic with respect to the axis. As soon as it reaches (infinitesimally close to) a vertex, we declare that vertex to be v_j. If it never encounters an edge or a vertex, then it will exit at the right end of the cylinder. In this case we declare v_n to be the desired v_j. The movement of the particle ensures that the edge (v_i, v_j) can be added preserving cylindricality. A similar procedure applied after this will work to make all sources other than v_1 have incoming edges. □

Theorem 3.4. *Let G be any planar directed acyclic graph. The following are equivalent.*

(i) G has a cylindrical embedding.

(ii) G is a spanning subgraph of a cylindrical SSPD.

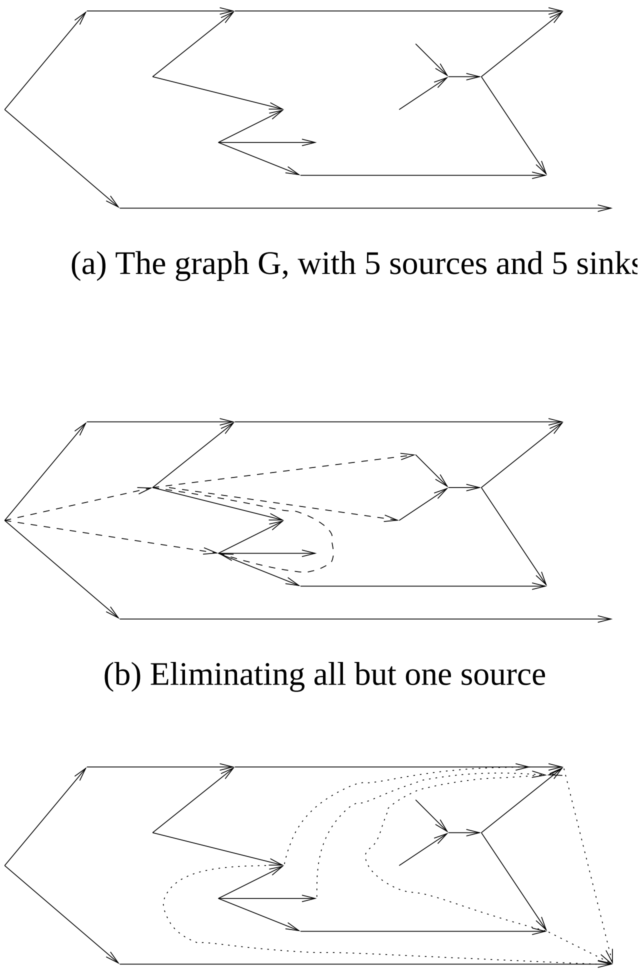
(iii) G is a spanning subgraph of an SSPD.

It follows that testing for cylindricality is in NP. However, though cylindricality generalizes upward planarity, testing for which is NP-complete, it is possible that testing for cylindricality is easier.

One direction of the theorem above is already constructive using Lemma 3.2. We make the proof of Lemma 3.3 constructive via a more complicated construction. This construction works only for one stage (multiple sinks to single sink or multiple sources to single source), and yields only a planar (not cylindrical) embedding of H. The advantage is that it is implementable in logspace.

Lemma 3.5. *Let G be a connected (in the undirected sense) cylindrical DAG with S sources and T sinks. Given the cylindrical embedding of G, we can construct, in L, a planar single-source DAG H_s with T sinks and a planar single-sink DAG H_t with S sources such that G is a spanning subgraph of both.*

Proof. We describe how to construct H_s; the construction of H_t is symmetric. Since G is connected, for every face f, the edges incident on f form a connected graph. For each face f, let i be the smallest index such that v_i is on the boundary of the face. Then there is some edge $e = (v_i, v_j)$ such that $f = R(e)$. Start traversing the boundary of f,



(a) The graph G, with 5 sources and 5 sinks

(b) Eliminating all but one source

(c) Eliminating all but one sink

Figure 3.2: Obtaining H from a connected G.

starting with such an edge $e = (v_i, v_j)$. For each v_k encountered on the boundary with in-degree 0, add edge (v_i, v_k). See Figure 3.2 (a), (b) for an example.

Clearly this preserves acyclicity, since all new edges are from a lower indexed to larger indexed vertex. This also preserves planarity. The new edges are inserted, in the order encountered, into the cyclic ordering around v_i immediately after the arc (v_i, v_j). A new edge (v_i, v_k) is inserted into the cyclic ordering around v_k immediately after the arc (v_l, v_k) which led to the discovery of v_k on this face boundary. Thus we can easily compute the new planar combinatorial embedding.

As the figure shows, we may end up adding far more edges than is necessary. (Multiple edges will not get added if we process each face sequentially. But in logspace, we cannot cascade polynomially many such stages. So while processing each face, we check for in-degree zero in the original graph.) Since G is connected, every source has a path to v_1. Hence every source lies on the boundary of at least one face with a lower indexed vertex, and hence acquires an incoming edge. Thus at the end, only v_1 is a source. □

As figure 3.2 (b) shows, applying the above construction on a graph to remove multiple sources may trap a sink. So we cannot sequentially remove multiple sources and then multiple sinks. In fact, after removing multiple sources, we do not know if the graph H so constructed necessarily has a cylindrical embedding. Even if it does, we do not know how to recover one.

In the above proof, connectedness ensured that every source other than v_1 acquired an incoming edge. We observe in the following lemma that absolute connectedness is not a critical requirement.

Lemma 3.6. *Let G be a cylindrical DAG where each connected component of the underlying undirected graph has either a single source or a single sink. Then a planar single-source single-sink DAG H of which G is a spanning subgraph can be constructed in* L.

Proof. Partition the underlying undirected graph of G, in L, into connected components $G_1, \ldots, G_c$. For each component, there is a cylindrical embedding inherited from that of G, which can be efficiently retrieved. By Lemma 3.5, each G_i is a spanning subgraph of a planar DAG H_i, with a single-source s_i and single-sink t_i, and H_i can be constructed in logspace. All that remains is to combine these H_i. Since each H_i is acyclic, the graph H obtained by adding edges t_i, s_{i+1} is also acyclic, and has a single source s_1 and single sink t_c. To see why it is planar, consider planar embeddings of each H_i with s_i on the external face. (The construction of Lemma 3.5 does yield such embeddings.) Consider any face f for which t_i is on the boundary. We insert the embedding of H_{i+1} in this face, and connect t_i to s_{i+1}. (See Figure 3.3.)

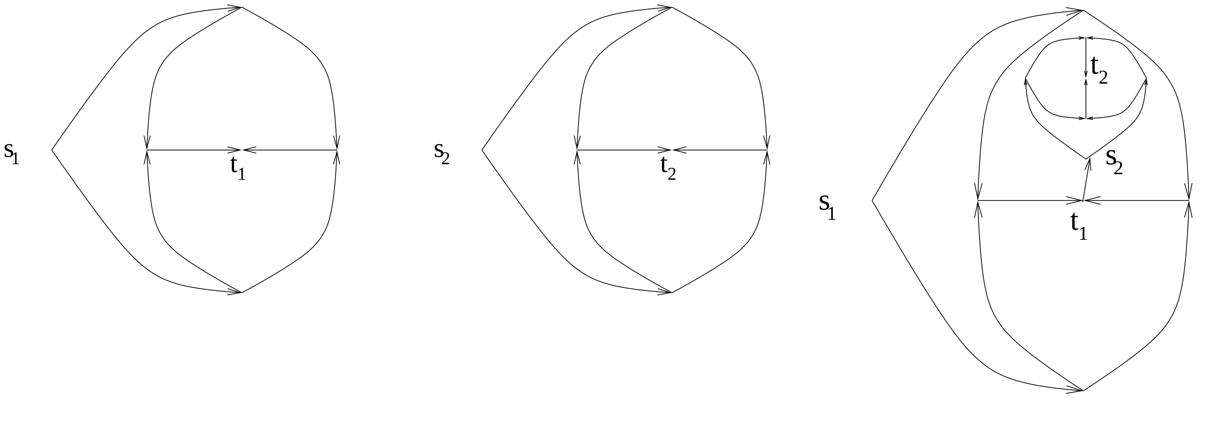


Figure 3.3: Patching H_1 and H_2 preserving planarity.

To construct a planar embedding of H, we can simply construct afresh a planar embedding of H in L. (Strictly speaking, this is not necessary. The edge (t_i, s_{i+1}) can be inserted anywhere in the cyclic ordering of t_i. In the cyclic ordering of s_{i+1}, it should be inserted in such a way that it lies on the external face of H_{i+1}. Given the way H_{i+1} is constructed from G_{i+1}, this information about the external face is indeed available.) □

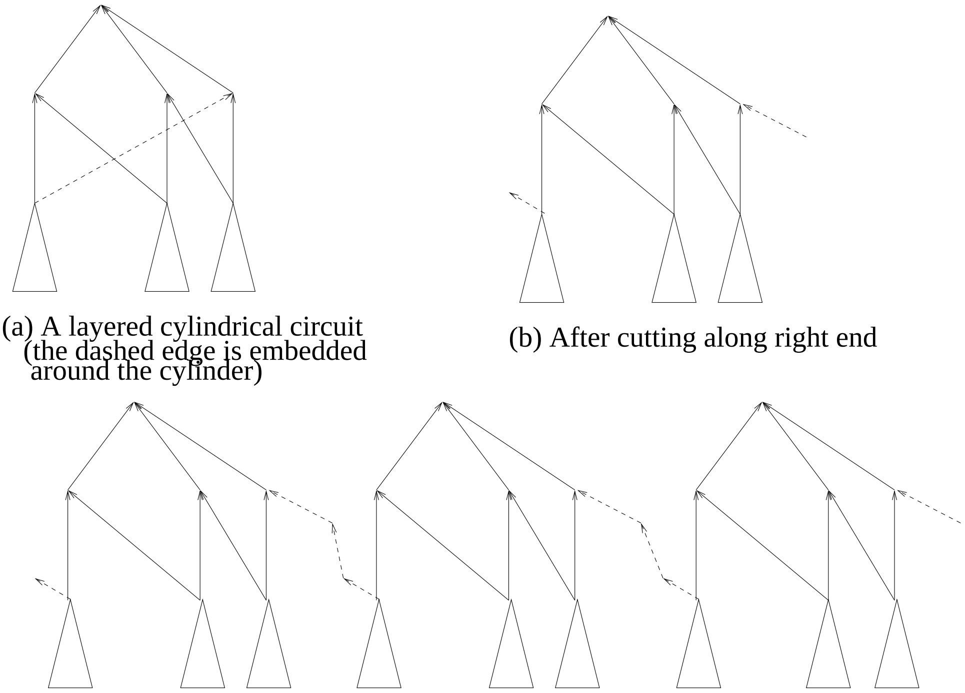
4. Circuits on cylinders

We now show that for circuit evaluation, any technique applicable to layered upward planar circuits also applies to cylindrical circuits, with a uniformity requirement in

$\mathsf{L}(\mathsf{PDLP}) \subseteq \mathsf{L}(\mathsf{NL})$. The result is obtained in two stages: first we show how to deal with layered cylindrical circuits, and then we show how to layer arbitrary cylindrical circuits. We also show that one-input-face circuits reduce to upward stratified circuits, with a similar uniformity requirement.

Lemma 4.1. *Given a circuit C with a layered cylindrical embedding $\mathcal{E}$, we can in logspace obtain an equivalent circuit C' with a layered upward planar embedding $\mathcal{E}'$. Further, if $\mathcal{E}$ is stratified, so is $\mathcal{E}'$. Also, if C is monotone, so is C'.*

Proof. Intuitively, what we want to do is as follows. Consider a geometric embedding of C on the plane, with layers corresponding to concentric circles and edges travelling inwards. By rotating a ray shooting out of the root, we can find an angular position where it does not contain the embedding of any node. By deforming edge representations if necessary, we can ensure that each edge intersects the ray (at this angular position) in at most one point. Now simply "cut" the circuit C along the ray. This gives rise to dangling in-edges and out-edges and a circuit D which is layered upward planar. Patch multiple copies of D side-by-side, feeding zeroes to the dangling edges of the extremal copies, and let the root of the middle copy be the new root. See Figure 4.1.



(a) A layered cylindrical circuit (the dashed edge is embedded around the cylinder)

(b) After cutting along right end

(c) Joining up copies of cut circuit

Figure 4.1: Obtaining an upward circuit equivalent to a cylindrical one.

To translate this into a formal proof, we need to describe (a) how to obtain, in logspace, the curve along which we will cut the circuit C to get D, (b) how the copies will be patched functionally, (c) how the embeddings of the copies will be patched, and (d) why the resulting circuit is equivalent to C.

We first perform some preprocessing on the circuit. Since we are given the layering as well as the label r of the circuit output gate, we can throw away all gates at a larger layer than r. Now, treating all edges as undirected, use the logspace connectivity algorithm to delete all gates with no (undirected) path to r. Let the resulting circuit be C_1, with layers $V_0, V_1, \ldots, V_h$ and r at layer h. We replace each vertex u by vertices u_{in} and u_{out} with a directed edge from u_{in} to u_{out}. The type of gate u_{in} is the type of u, while u_{out} is a NO-OP gate. An edge (u, v) is replaced by the edge (u_{out}, v_{in}). The resulting circuit, call it C_2, has $2h$ layers: an out layer for V_0, an in layer for V_h, and two for all other layers. The layered cylindrical embedding of C_2 is easily obtained from that of C_1, and hence of C, in logspace. The only tricky point is handling sources/sinks of C_1. If u is a source of C_1, we need to decide where to insert the edge (u_{in}, u_{out}) into the cyclic ordering of edges leaving u_{out}. This is where we need the third part of the representation of cylindrical embeddings: we insert this edge just before the edge $L(u)$. Similarly for a sink v, we insert (v_{in}, v_{out}) in the ordering around v_{in} just after $L(v)$. C_2 is clearly equivalent to C; further, it has the nice property that no layer has a source as well as a sink.

To see (a), we start with vertex r of C_2. At some stage, suppose that the path ρ under construction has reached vertex g from above. If g is at the lowest layer, we are done. Otherwise, move down to any neighbour of g at a lower layer. Suppose there is no such neighbour; that is, g is a source node. Then g is of the form v_{in} for some $v \in C_1$. Traverse the boundary of the face to the right of (v_{in}, v_{out}), until it first encounters a vertex g' at a layer lower than g. Such a vertex must exist, since g has undirected connectivity to r which has undirected connectivity to the layer below g. The path ρ now proceeds from g to g'.

The path ρ constructed uses some circuit edges and some dummy edges. Let C_3 be the graph $C_2 \cup \rho$. The above procedure of constructing ρ also gives us a layered cylindrical embedding of C_3.

We cut C_3 to the immediate right of the path, starting at r, to obtain a layered upward planar circuit C_4. The embedding of C_4 is specified as follows: Retain edges (u, v) where neither u nor v is on ρ. For u on ρ, retain edges leaving or entering u to/from the left of ρ. Replace an edge (u, v) leaving ρ on its right by the edges (x, v) and (u, x'), where x and x' are new gates of fan-in/fan-out zero. Similarly, replace an edge (w, u) entering ρ from its right by the edges (y', u) and (w, y), where y' and y are new gates of fan-in/fan-out zero. It is clear that this can be performed in logspace. C_4 is the circuit D informally described earlier.

Let d be the depth of C_2. Place $2d+1$ copies of C_4 side by side in a row. Identify new node x' of copy i with new node x of copy $i+1$. Identifying x and x' gives a subdivision of an edge present in a copy of C_4. Restore the subdivision to a single edge (remove the identified node). New nodes x of the leftmost copy, and new nodes x' of the rightmost copy, are fed constant 0, via paths of NO-OP gates of appropriate length (this is done to preserve stratifiedness). See Figure 4.1 (c). Designate the root of copy $d+1$ as the new root. Let this circuit be called D. It is easy to see that D is layered upward planar, and that its embedding can be obtained from that of C_4 in logspace. Also, if C_2 is stratified, so are C_4 and D.

We claim that D is equivalent to C_2, and hence to C. The reason is simple: at the lowest level, all nodes of D are correct (they evaluate to the same value as corresponding nodes in C_2). If at level l, the copies $i-1, i, i+1$ of C_4 are correct, then at level $l+1$

the ith copy of C_4 is correct. Thus over $2d+1$ levels, we may lose at most $2d$ copies, but the central copy will correctly evaluate the root of C_2. □

In the above proof, the layering of the given circuit appears crucial. We observe below that without layering, the same conversion can be performed in $\mathsf{L}(\mathsf{PDLP})$.

Lemma 4.2. *Evaluating a circuit C with a cylindrical embedding $\mathcal{E}$ reduces in* $\mathsf{L}(\mathsf{PDLP})$ *to evaluating a layered cylindrical circuit C' with embedding $\mathcal{E}$'. Further, if $\mathcal{E}$ is one-input-face, then $\mathcal{E}$' is stratified. Also, if C is monotone, so is C'.*

Proof. We proceed in four steps.

1. We remove from C all nodes with no directed path to the output gate of C. This gives an equivalent circuit G with a single sink, and with an inherited cylindrical embedding.

2. From the given cylindrical embedding of G, we construct the SSPD H with the same vertices as G and containing all the edges of G.

3. Using the layered embedding algorithm of Figure 3.1, we obtain a layered cylindrical embedding of an SSPD H', obtained by subdividing edges of H into directed paths.

4. We recover a layered cylindrical embedding of a digraph G' from that of H' by simply throwing away all directed paths corresponding to edges in $H \setminus G$. We convert G' to a circuit by specifying that all the new subdivision vertices have type NO-OP.

Since C is a planar DAG, Step (1) can be performed in $\mathsf{L}(\mathsf{PDLP})$. Step (2) uses Lemma 3.5, and can be performed in logspace. Step (3) uses Lemma 3.2, and runs in $\mathsf{L}(\mathsf{PDLP})$. It is straightforward to see that Step (4) can be performed in logspace. □

Note that the layered embedding algorithm needs a single-sink one-input-face embedding. In the above proof, the one-input-face condition is achieved in step 2 by exploiting cylindricality. However, if the given circuit already has a one-input-face embedding, then cylindricality is not needed. Thus we have:

Lemma 4.3. *Evaluating a circuit C with a one-input-face embedding $\mathcal{E}$ is reducible, in* $\mathsf{L}(\mathsf{PDLP})$, *to evaluating a stratified cylindrical circuit C' with embedding $\mathcal{E}$'. Also, if C is monotone, so is C'.* □

5. Improved Upper bounds for MPCVP

In this section we revisit some of the MPCVP algorithms in the literature. We observe that some of these algorithms have tighter bounds than claimed. Wherever possible, we apply (some of) the reduction lemmas of Section 4 to expand the class of circuits for which the algorithm applies. Wherever possible, we also try to weaken the input requirements.

Goldschlager [Gol80] considered upward stratified circuits. He showed that in this special case, if the corresponding embedding is given with the input, then MPCVP is in NC^2. This upper bound was improved to LogCFL by Dymond and Cook [DC89].

They use the characterization (due to [Coo71, Sud78]) of LogCFL as languages accepted by polynomial-time-bounded pushdown automata augmented with an auxiliary logspace worktape, AuxPDA(poly) in short. (Similarly, $\mathsf{LogDCFL}$ is characterized as languages accepted by deterministic polynomial-time-bounded pushdown automata augmented with an auxiliary logspace worktape, DAuxPDA(poly).)

The main idea behind obtaining the LogCFL bound is as follows: since the circuit is monotone, intervals of contiguous 1s at the input level travel upwards as contiguous segments which may shrink, expand, or merge, but never split. (This last property breaks down if the embedding is not stratified.) So evaluating the given circuit C amounts to proving that an interval is true (or valid), by finding a set of intervals at the previous level which imply its validity, and recursively proving their validity. An important property of a minimal set of intervals proving validity of the root (a "proof tree" on intervals) is that it is polynomial sized; hence an auxiliary push-down automaton performing the recursive verification nondeterministically will run in polynomial time. But this is precisely the class LogCFL.

The work of Barrington *et al.*[BLMS99] brings the evaluation of monotone upward stratified circuits, presented along with such an embedding, down to $\mathsf{LogDCFL}$ by evaluating the circuit in a bottom up fashion. The DAuxPDA algorithm repeatedly transforms the input by (a) detecting when a 0- or 1- interval at the input layer fails to propagate high enough, and (b) replacing the interval by all 1s or all 0s. The transformation thus preserves the value of the output gate. The stack is used to keep track of the frontier up to which simplifying transformations have been made. Polynomial running time is ensured, amongst other things, by the placement of a virtual blocking interval of 0s on either extreme at each level. The algorithm requires the upward stratified embedding to be supplied as input. Though not stated explicitly, it also works for circuits with multiple sinks. (The only point to be checked is that intervals of 1s may merge though separated not just by a 0 interval but by 0- and 1- intervals, all arising at sinks; see the discussion preceding Proposition 8 of [BLMS99]. This makes no difference to the technical claims.)

Since virtual blocking intervals cannot be placed at extremes of each layer for a cylindrical embedding, we do not see how to extend this algorithm to work for stratified cylindrical circuits. However, we can still obtain this upper bound by using Lemma 4.1 in conjunction with this algorithm:

Theorem 5.1. *Given a monotone planar circuit C with a stratified cylindrical embedding, determining whether C evaluates to 1 is in* $\mathsf{LogDCFL}$. □

What if the embedding needed for Theorem 5.1 is not explicitly given, but there is the promise that such an embedding exists? At some cost, we can recover a suitable embedding. The cost is high enough that we can weaken the premise further. Note that stratified cylindrical embeddings are one-input-face, though the converse may not hold. But one-input-face embeddings can be constructed in logspace. With such an embedding, we can apply Lemma 4.3 and Theorem 5.1; thus we get a slightly weaker upper bound for a more general class:

Theorem 5.2. *Given a monotone planar circuit C, if C has a one-input-face embedding, then C can be evaluated in* $\mathsf{L}(\mathsf{PDLP} \oplus \mathsf{LogDCFL}) \subseteq \mathsf{L}(\mathsf{NL} \oplus \mathsf{LogDCFL}) \subseteq \mathsf{LogCFL}$.

Proof. We first construct a one-input-face embedding of C in logspace, as described in Section 2.2. Then we apply Lemma 4.3 to obtain an equivalent cylindrical stratified circuit C', and use Theorem 5.1. □

Layered one-input-face circuits were considered by Yang [Yan91] as a step towards placing general MPCVP in NC. Note that these are precisely cylindrical stratified circuits. In Section 2 of [Yan91], an upper bound of NC^2 is obtained for evaluating such circuits. Rather than use a tool like Lemma 4.1 followed by the bound of [Gol80], Yang devised a somewhat different algorithm, since a modification of it was used in a later section. The essence of his algorithm was the same as in [DC89]: evaluating the given circuit C is equivalent to evaluating a circuit C' which tries to determine, for each interval or segment of gates at each level, whether this interval evaluates to all 1s. Further, he carried the range of inputs used in proving validity as a parameter. That is, for each interval i, j of gates numbered between i and j at level l, and for each input range x, y, determine if the interval i, j, l can be proved valid using only inputs from the range x, y. (Note: it is not claimed that all inputs in the range x, y are 1s, merely that 1s outside this range are not needed for proving validity.) By doing this, he was able to establish that C' has polynomial algebraic degree. Then he appealed to [MRK88] to obtain the NC^2 bound. However, it is now known that circuits of degree polynomial in circuit size can be evaluated in LogCFL [Ruz80, Ven91]. Thus we have

Proposition 5.3. *The algorithm of Section 2 from [Yan91], for evaluating instances of MPCVP presented with cylindrical stratified embeddings, has a* LogCFL *implementation.* □

Another notable point is that though Yang assumed a single-sink circuit, his algorithm works also in the presence of multiple sinks.

This bound was independently obtained by Delcher and Kosaraju [DK95], who observed that the algorithm of [DC89], though presented only for upward stratified circuits, works also for the cylindrical stratified case. This is because even for such embeddings, the proving sub-circuit for validity of intervals has a tree structure which is polynomial-sized.

In [Kos90], the requirement that the circuit be stratified was dropped for the first time. The input is required to be a monotone layered upward planar circuit, with the witnessing embedding supplied. Dropping the stratified (one-input-face, for layered circuits) condition means that intervals of contiguous 1s can split due to the presence of an input node at an intermediate layer, and this makes all the preceding algorithms for upward-planar or cylindrical stratified circuits inapplicable. Kosaraju's idea is, however, quite simple and elegant: repeatedly split the circuit horizontally at a layer such that both pieces are between 1/4 and 3/4 of the entire circuit in size. Evaluate each piece recursively, replacing cut off wires by variables. (The details of the recursive splitting are a bit sketchy in [Kos90] but are supplied in full in [DK95] for the stratified case.)

But what does it mean to evaluate a circuit with variables? Due to monotonicity, if a gate evaluates to 1 (0) even when all variables are set to 0 (1, respectively), then the gate evaluates to 1 (0, respectively) for all settings of the variables. So by evaluating such a circuit on two settings — all variables 1, and all variables 0 — the gates can be partitioned into three sets: evaluating to 1, or 0, or depending on the input variables. Once the recursive evaluation is done, the bottom piece is entirely evaluated and the top piece has some variable gates. But now the values of all its variable inputs are known from the bottom piece, so this piece can be fully evaluated.

Clearly, the recursion depth is logarithmic, and the base case of recursion is a monotone upward stratified circuit with variables. As observed above, [Kos90] used the fact that the NC^2 bound of [Gol80] applies also in the presence of variables to obtain the three-part partition. Using this bound for the base case, [Kos90] reported an upper bound of NC^3.

It is worthwhile noting that at internal stages of the recursion, the circuits could become generalized; they could have constant gates with non-zero fan-in (e.g. an OR gate could get as inputs one 1 and one variable from the preceding level of recursion). So, to apply Goldschlager's algorithm to the base case, the constant gates with non-zero fan-in are explicitly removed. That is, to patch up the two pieces, only the sub-circuit induced by gates which depend on variables is considered.

It is also worthwhile noting that this algorithm is also insensitive to multiple sinks, since the strategy evaluates not just a designated sink but every gate in the circuit.

Kosaraju's upper bound can be tightened by noting that a log-recursion-depth algorithm, using the algorithm of [BLMS99] rather than [Gol80] for the base case, yields an implementation in $\mathsf{AC}^1(\mathsf{LogDCFL})$.

Proposition 5.4. *The algorithm of [Kos90], for evaluating instances of MPCVP presented with layered upward planar embeddings, has an* $\mathsf{AC}^1(\mathsf{LogDCFL})$ *implementation.* □

Further, the class of circuits for which this bound applies can be expanded to cylindrical circuits:

Theorem 5.5. *An instance of MPCVP, presented with a cylindrical embedding, can be solved in* $\mathsf{AC}^1(\mathsf{LogDCFL})$. □

Proof. Let C be the given circuit with a cylindrical embedding. Using Lemma 4.2, we obtain in $\mathsf{L}(\mathsf{PDLP}) \subseteq \mathsf{NL} \subseteq \mathsf{AC}^1$ an equivalent circuit C' with a layered cylindrical embedding $\mathcal{E}$. Applying Lemma 4.1 gives, in $\mathsf{L} \subseteq \mathsf{AC}^1$, an equivalent layered upward planar circuit C'', to which the preceding proposition can be applied. Note that for subcircuits evaluated at recursive steps, embeddings are inherited from $\mathcal{E}$. □

Bi-cylindrical Circuits : We now consider a generalization of cylindrical circuits, which we call *bi-cylindrical* circuits. These strictly subsume cylindrical, while still lying within planar circuits.

Definition 5.6 (Bi-cylindrical circuits). *A DAG or circuit G is bi-cylindrical if it has an embedding on the surface of the cylinder such that there is a circle C going around the cylinder surface, and all edges go towards C.*

Thus C splits G into two pieces (overlapping only on C) where each piece is cylindrical. (See Figure 5.1.)

Now each piece can be evaluated separately, and the the root gate can then be evaluated from its values in the two pieces. Depending on whether the pieces are layered or not, and whether they have one-input-face embeddings or not (if both do, then all inputs lie on the two extreme ends of the bi-cylinder), we have the following upper bounds:

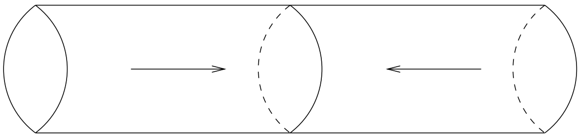


Figure 5.1: Bi-cylindrical embeddings

bi-cylindrical circuit type	layered	not layered
inputs only at extremes	LogDCFL	$\mathsf{L}(\mathsf{PDLP} \oplus \mathsf{LogDCFL})$
inputs anywhere	$\mathsf{AC}^1(\mathsf{LogDCFL})$	$\mathsf{AC}^1(\mathsf{LogDCFL})$

Focused circuits : Focused embeddings are considered in [DK95], since they arise in recursive stages of their final algorithm for general MPCVP. Recall that a focused embedding is one where all sources other than those in a designated face f feed into a node reachable from a source in f. This is a topological analogue of a skewness condition on circuits. Such a circuit C can be converted to an equivalent upward stratified one C' (with such an embedding explicitly obtained) by simplifying the neighbours of the inputs not on the special face and then using Lemma 4.3 followed by Lemma 4.1. One consequence is that some internal nodes may be constant nodes; e.g. an OR gate with a skew 1 input from outside f simplifies to a constant gate, but still has another input wire feeding into it. We could cut off such wires as well. (But we must do this *after* obtaining the stratified cylindrical embedding; if we do it before that, then the resulting circuit is no longer one-input-face, so Lemma 4.3 does not apply.) After this cutting, the resulting circuit C'' won't be stratified, so we can only use the bound of Theorem 5.5 and not that of Theorem 5.1. Since C' can be obtained from C in $\mathsf{L}(\mathsf{PDLP}) \subseteq \mathsf{AC}^1$, and since C'' can be obtained from C in logspace, we have:

THEOREM 5.7. *Given a monotone planar circuit C with a focused embedding, determining whether C evaluates to 1 is in* $\mathsf{AC}^1(\mathsf{LogDCFL})$. □

The final algorithms of both [Yan91] and [DK95] make no assumptions about the embedding; given an instance of MPCVP with *any* planar embedding, they show that evaluation is in NC. Both algorithms repeatedly evaluate carefully chosen smaller circuits with special embeddings (cylindrical stratified or focused). But the noteworthy point is that these special embeddings for the smaller circuits can always be obtained, in NC, from the given planar embedding.

Yang's analysis proceeds by showing that $O(\log n)$ iterations of the following suffice: For each face f containing some inputs, consider the subcircuit C_f reachable (in a directed sense) from f. C_f can have some dangling in-edges from the rest of the circuit; replace these by variables to get a circuit with variables and a focused embedding. Evaluate this circuit as far as possible (the variables, or unknown wires, do not allow complete evaluation), using a generalization of the scheme leading to Proposition 5.3. Then perform some obvious simplifications, and reiterate.

The generalization does not permit the use of [BLMS99] or Theorem 5.1. However, the strategy is the same as originally used by Yang for one-input-face embeddings; namely, there is an equivalent polynomial degree circuit doing this partial evaluation. Hence, by [Ven91], it can be performed in LogCFL. Hence, a careful analysis of Yang's algorithm allows us to conclude that MPCVP is in $\mathsf{AC}^1(\mathsf{LogCFL})$. However, it can be seen that this class is the same as SAC^2. Thus we have the following:

THEOREM 5.8. *Given a monotone planar circuit C, determining whether C evaluates to 1 is in* SAC^2. □

6. Extensions

In this section we extend the ideas developed in the previous sections to some non-planar cases, as well as to the non-monotone case with limited negations.

6.1. Monotone circuits on the torus. We start with the case of a torus which is the canonical surface of genus 1. A digraph is *toroidal* if it can be embedded on a torus. We look at circuits whose underlying DAG is toroidal. We assume that the toroidal embedding is given as a combinatorial embedding; verifying that this embedding has genus one can be done in logspace.

Any closed curve separates the plane into disconnected regions, but a closed curve can disconnect the surface of a torus or leave it connected. In the latter case, it is called a surface non-separating curve. Any non-planar toroidal graph has at least one surface non-separating cycle. The following lemma is from [ADR05a]:

LEMMA 6.1 ([ADR05a]). *Given a non-planar graph G with an embedding on the torus, a surface non-separating cycle in G can be found in L.*

Using this result, we establish the following reduction lemma, which along with Theorem 5.8, immediately gives the main result of this section.

LEMMA 6.2. *A circuit C with a toroidal embedding can be converted in log space to an equivalent circuit C' with a planar embedding. Also, if C is monotone, so is C'.*

PROOF. The lemma is proved by essentially using the idea from [ADR05a]. Intuitively what we want to do is as follows. Consider a given toroidal embedding. Using Lemma 6.1, we will find a cycle (in the undirected sense) such that "cutting" the circuit along the cycle will make the remaining graph planar. Now we will paste together several copies as in the cylindrical case such that one copy evaluates to the same function as the original circuit. Also, the pasting will be done preserving planarity.

As in Lemma 4.1, to translate this into a formal proof, we need to describe (a) how to obtain, in logspace, the curve along which we will cut the circuit (b) how the embeddings of the copies will be patched, (c) how the copies will be patched functionally, and (d) why the resulting circuit is equivalent to C.

For (a) and (b), we use Lemma 6.1. Borrowing the notation from [ADR05a], let $v_1, v_2 \dots v_r$ be the non-separating cycle returned by the logspace procedure. Let G' be the graph obtained after cutting along this cycle. This graph will have two copies of the vertices on the cycle on each end of the cylinder. Let these be $v_{1,1}, v_{2,1}, \dots v_{r,1}$ and $v_{1,2}, v_{2,2}, \dots v_{r,2}$ respectively. Let d be the depth of the original circuit, we make $2d+1$ copies of the circuit and place them side by side, identifying the corresponding vertices and edges. The combinatorial embedding of C' is obtained exactly as in Section 3 of [ADR05a], see Figure 6.1 for an illustration. Clearly, C' is planar, since it has an embedding on the surface of the cylinder. (Note, however, that the embedding may not be "cylindrical".)

For (c), each gate in each copy behaves exactly as in the original circuit. Edges coming into the extreme copies from outside are set to source nodes with value 0. Let this new circuit be called C'.

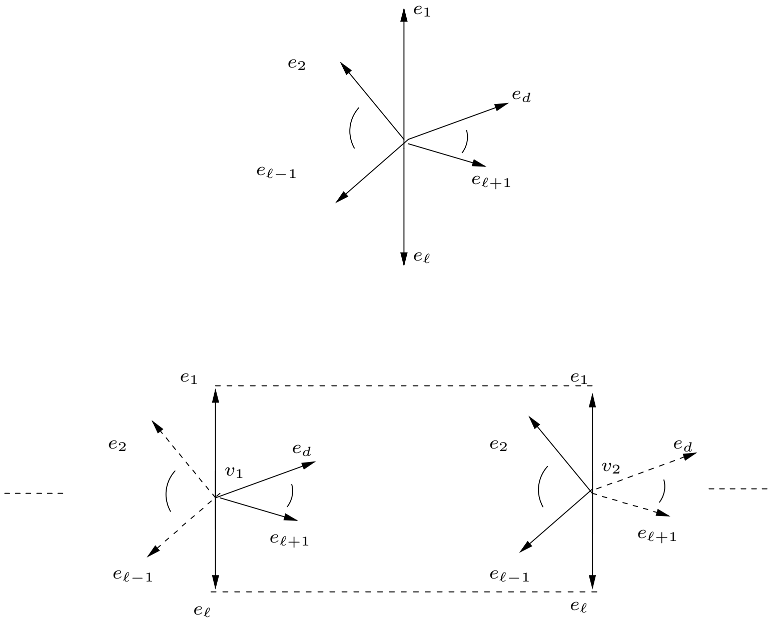


Figure 6.1: Patching the copies

Now to argue (d), we introduce the notion of *cycle-height*. Let c be the non-separating cycle with respect to which cutting has been performed. The cycle-height of gate g is the smallest non-negative integer k such that every path from a leaf to g "crosses" the cycle c at most k times. By a simple inductive argument, we can establish that if gate g has cycle-height k, then all copies of g in C', except those in the leftmost k and rightmost k copies of C, evaluate to the same value as g in C. If follows that in the middle copy, all the gates will get evaluated correctly. □

THEOREM 6.3. *A monotone circuit, given with an embedding on a torus, can be evaluated in* SAC^2. □

An obvious question is whether the above technique can be extended to give an NC upper bound for higher genus circuits. The limitation is that if we do not get a genus 0 surface to make copies, then the process of making copies will increase the genus.

6.2. Monotone multi-cylindrical circuits. We extend the idea of bi-cylindrical circuits in a natural way to what we call *multi-cylindrical* circuits. Such circuits strictly subsume the bi-cylindrical case, but are incomparable with planar circuits. A noteworthy point is that a multi-cylindrical circuit can be of arbitrary genus.

A *k-cylindrical* circuit can be presented as a set of k components. Each of these has a cylindrical embedding. The edges of each cylindrical component flow towards the right rim. And the right rims of each can be identified (let us call that curve c). Another circuit sits on the gates in c such that all the inputs to this circuit come only from gates in c. This circuit is also cylindrical.

A *multi-cylindrical* circuit is a k-cylindrical circuit, for some k.

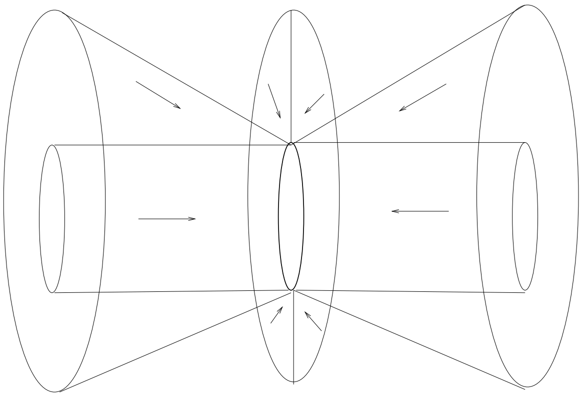


Figure 6.2: Multi-cylindrical embeddings

Notice that 2-cylindrical according to this definition is stronger than the bi-cylindricality discussed earlier. This is because we allow a circuit C' sitting on the nodes on c. But allowing this is also essential, since each gate is assumed to have fan-in at most 2. If such a construct were not allowed, then the root gate would itself have to sit on c and take inputs from at most 2 components. The other components would play no role at all and could be excised, making k-cylindrical equal to 2-cylindrical for $k > 2$. On the other hand, allowing such a construct, bi-cylindrical circuits are exactly those 2-cylindrical circuits for which C' is the trivial circuit; it merely pulls out the value of a fixed gate appearing on the curve on c.

Let C' be the cylindrical subcircuit sitting over the nodes in c. Now C' can be thought of as a circuit which has c as its set of input nodes. We can evaluate each of the cylindrical components separately in parallel. With this, we get the value of each node in c. Now we can evaluate C' using the values of nodes in c. In fact, for upper bounds in NC, we don't even require C' to be cylindrical; it can be planar or toroidal as well. Depending on the complexity of evaluating each component, and of evaluating C' from c, we have the following upper bounds:

Inputs on c_i's	Type of C'	layered	not layered
only at extremes	cylindrical stratified	LogDCFL	L(PDLP $\oplus$ LogDCFL)
anywhere	cylindrical	AC^1(LogDCFL)	AC^1(LogDCFL)
anywhere	planar	AC^1(LogDCFL)	SAC^2
anywhere	toroidal	–	SAC^2

As one can see, this gives upper bounds only for the promise problem. Also, one limitation is that we do not know the complexity of obtaining such an embedding if one exists, and hence the embedding need to be explicitly given along with the input. As far as we know, this is the first result on evaluating a class of monotone circuits which contains some arbitrary genus circuits, in NC. Clearly, if P $\neq$ NC, there are high genus circuits which do not have multi-cylindrical embeddings.

6.3. Circuits with limited negations. We now consider planar circuits which are not monotone, but where the negations gates are limited in some way. Without such a limitation, there is no hope of evaluating the circuit inside NC unless P=NC, since planar CVP is known to be P-complete [Gol77]. How many negation gates are needed to obtain this hardness? We show in this section that unless P=NC, there are P-computable functions requiring super-polylogarithmic number of negation gates in any poly-sized planar (and even toroidal) circuit computing them (Lemmas 6.4,6.5).

Markov [Mar58] came up with a surprisingly tight bound on the number of negation gates that are needed to compute any boolean circuit. He showed that to compute a boolean function on n variables, $\lceil \log(n+1) \rceil$ negation gates are necessary and sufficient. One natural question to ask is whether such a bound holds for restricted families of circuits as well. Fischer [Fis74] showed that for every poly-sized log-depth circuit, there is another equivalent poly-sized log-depth circuit which uses at most $\lceil \log(n+1) \rceil$ negations. A noteworthy point in Fischer's construction [Fis74] is that it is not planar; so it does not imply that evaluating planar circuits with $O(\log n)$ negations is P-hard. In contrast, Santha and Wilson [SW91] showed that there are functions requiring super-logarithmic number of negation gates in any poly-sized constant-depth circuit computing them. Our result can be viewed as a conditional topological analogue of this result, restricted to P-computable functions.

Let us try to evaluate a non-monotone planar circuit in parallel. The computation proceeds in stages. For any gate g where the subcircuit rooted at g has no negations, the value of g can be found in SAC^2, by Theorem 5.8. Assume that all such gates have been evaluated. Now let g be a gate such that in the sub-circuit rooted at g, a root-to-leaf path has at most one negation gate. Such gates can be evaluated by an SAC^2 circuit whose inputs include the original circuit input, the values of the gates already evaluated, and the negations of these values. Generalizing this, we define *negation-height*, akin to the notion of cycle-height from the proof of Lemma 6.2. The negation-height of an input gate (variable or constant) is 0, by convention. The negation-height of gate g is the smallest non-negative integer h such that every path from a leaf to g has at most h negation gates. At stage k, we evaluate all gates at negation-height k. The inputs to the stage-k circuit are the circuit inputs, and the values as well as negated values of all gates at negation-height $j < k$. Each stage k has an SAC^2 circuit, obtained by putting together the SAC^2 circuits for each gate at negation-height $j < k$. Thus if gate g has negation-height k, then g can be evaluated by a polynomial-sized semi-unbounded circuit of depth $O(k \log^2 n)$.

Of course, this requires negation-height to be explicitly available. By placing a weight of 1 on edges out of a negation edge, and a weight of 0 on other edges, we see that negation-height of g is exactly the maximum weight of a g-to-leaf path. Since the circuit is a DAG, this is in NL, in SAC^2. So computing the negation-height is not a real bottleneck.

We thus have the following result:

LEMMA 6.4. *A planar circuit in which the output gate is at negation-height k can be evaluated by a polynomial size semi-unbounded circuit of depth $O(k \log^2 n)$. Thus planar circuits with polylog negation-height can be evaluated in* NC. □

It is not necessary that the entire circuit be planar. Since the evaluation proceeds in stages, it is sufficient if for each h, the subgraph of all gates with negation-height h is planar. (It is easy to construct such circuits that are non-planar.)

Lemma 6.5. *A circuit C where*

(i) the output gate has negation-height k, and

(ii) for each $0 \leq h \leq k$, the subcircuit consisting of gates at negation-height exactly h is planar,

can be evaluated by a polynomial size semi-unbounded circuit of depth $O(k \log^2 n)$. □

This result can be combined with the results of Sections 6.1 and 6.2. If the (output gate of the) circuit has negation-height $k \in O(\log^i n)$, and if for each $0 \leq h \leq k$, the subgraph of gates with negation-height exactly h is toroidal or multi-cylindrical, then the whole circuit can be evaluated in NC, *provided* the appropriate embedding for each subgraph is given. (Such embeddings are not explicitly required in proving Lemma 6.5, since planar embeddings can be constructed in L.)

7. Discussion

This investigation leaves many questions unanswered.

1. Is cylindricality testing NP-hard? Recall that cylindricality strictly generalizes upward planarity, testing for which is NP-hard ([GT01]), and is strictly stronger than planarity, testing for which is in L([RR94, AM04, Rei05]). Actually, upward planarity testing becomes hard only in the presence of multiple sources, but is in AC^1 for single-source planar DAGs [BBMT98].

2. How can a cylindrical embedding be represented so that given a representation of this form, verifying if it is indeed cylindrical can be done in logspace? The representation we have used does not seem to have enough information for this.

3. Given a graph with the promise that it is cylindrical/layered cylindrical/layered upward planar, what is the complexity of recovering a witnessing embedding? This can make a big difference to the complexity of circuit evaluation; see item 6 below.

4. Let DLP_i denote the class of problems logspace many-one reducible to the problem *DAGLONGPATH* where the DAGs are unrestricted for $i = 0$, planar for $i = 1$, planar single-source or planar single-sink for $i = 2$, and planar single-source single-sink for $i = 3$. (Thus, DLP_1 is what is referred to as PDLP till now in this paper.) Let DR_i denote the class of problems logspace many-one reducible to reachability in the corresponding DAGs. Clearly, $\mathsf{DR}_i \subseteq \mathsf{DLP}_i$, and $\mathsf{DLP}_0 = \mathsf{DR}_0 = \mathsf{NL}$. What other relationships can be deduced among these classes?

   Notice that the layering algorithm of Figure 3.1 already needs a one-input-face single-sink planar DAG. A circuit on such a DAG can trivially be converted to an equivalent instance of DR_3 by adding a dummy source. Thus, the upper bounds of $\mathsf{L}(\mathsf{PDLP})$, obtained in Proposition 3.1 and Lemma 3.2, can actually be replaced by DLP_3, which may conceivably be stronger. In recent work by [ABC$^+$06], DR_3 and DR_2 are shown to be in L. Thus, if DLP_3 can be shown to be equivalent to DR_3, or reducible to DR_2, then the upper bounds of this paper will drop further. We need to be a bit careful: Lemma 4.2, for instance, uses Proposition 3.1 as well as DR_1 (step 1 uses DR_1 to obtain an equivalent instance of DR_2), and thus has a fine upper bound of $\mathsf{L}(\mathsf{DLP}_3 \oplus \mathsf{DR}_1)$. To establish Lemma 4.3, on the other hand, $\mathsf{L}(\mathsf{DLP}_3)$ suffices, since the first step is also dispensable. These finer bounds can be carried over to all the results of Section 5.

5. Recently, via a different approach bypassing Figure 3.1, Theorem 5.2 has been improved: one-input-face MPCVP has been shown to be reducible to layered upward planar monotone circuits, and hence is in $\mathsf{LogDCFL}$ [CD06]. It appears that focused MPCVP can also be captured in $\mathsf{LogDCFL}$ via this approach.

6. There are very few hardness results with respect to topological constraints. A well-known result due to [Bus87] says that evaluating a Boolean formula (the circuit is a tree) is NC^1-complete, thus MPCVP is at least NC^1-hard. A more recent notable result [Han06] shows that constant-width planar circuits characterize ACC^0. Are there natural topological restrictions which, placed on MPCVP, give instances complete for $\mathsf{LogDCFL}$, NL and LogCFL? In particular, is stratified cylindrical MPCVP hard for $\mathsf{LogDCFL}$?

   In [CD06], one-input-face MPCVP is shown to be hard for L. The hard instance produced here is in fact a width-2 tree. However, the result of [Han06] does not imply that evaluating it is in ACC^0, because the ACC^0 evaluation procedure of [Han06] explicitly needs the layered bounded-width presentation of the circuit, and it is computing this that is L-hard. Similarly, the result of [Bus87] does not imply that evaluating it is in NC^1, because the NC^1 evaluation procedure of [Bus87] requires the formula to be explicitly presented in fully parenthesized form, and computing this is L-hard. In other words, the hardness of evaluating one-input-face MPCVP lies in the hardness of obtaining a small-width specification, or even an explicit tree description, under the promise that the circuit is indeed a small-width tree. This situation thus underscores the difference that supplying an embedding can make; hence the importance of item 3.

   A special case of layered upward planar MPCVP arises when all AND gates are skew. (The hard instances of [CD06] are skew.) In this case, the circuit evaluates to 1 if and only if there is a path from an input labeled 1 to the root; it captures reachability in layered upward planar graphs. It is noteworthy that we do not know L-hardness for reachability in layered grid graphs, or even in grid graphs; the best lower bound is NC^1 (see [ABC$^+$06]). However, it is possible that layered upward planar monotone circuits are harder to evaluate than similar skew circuits.

Acknowledgements

A preliminary version of this paper appeared in the Proceedings of STACS 2006 LNCS vol. 3884 [LMS06]. The authors thank the anonymous referees of STACS 2006 and of this journal for useful comments.

References

[ABC$^+$06] E. Allender, D. A. Mix Barrington, T. Chakraborty, S. Datta, and S. Roy. Grid graph reachability problems. In *Proceedings of 21st IEEE Conference on Computational Complexity (CCC)*, pages 299–313, 2006.

[ADR05a] E. Allender, S. Datta, and S. Roy. The directed planar reachability problem. In *Proceedings of 25th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 3821 of *Lecture Notes in Computer Science (LNCS)*, pages 238–249, 2005.

[ADR05b] E. Allender, S. Datta, and S. Roy. Topology inside NC^1. In *Proceedings of 20th IEEE Conference on Computational Complexity (CCC)*, pages 298–307, 2005.

[AM04] E. Allender and M. Mahajan. The complexity of planarity testing. *Information and Computation*, 189(1):117–134, 2004.

[BBF03] C. Bachmaier, F.-J. Brandenburg, and M. Forster. Radial level planarity testing and embedding in linear time. In *Proceedings of 11th International Symposium on Graph Drawing (GD)*, volume 2912 of *Lecture Notes in Computer Science*, pages 393–405, 2003.

[BBMT98] P. Bertolazzi, G. Di Battista, C. Manning, and R. Tamassia. Optimal upward planarity testing of single-source digraphs. *SIAM Journal on Computing*, 27:132–169, 1998.

[BLMS99] D. A. Mix Barrington, C.-J. Lu, P. Bro Miltersen, and S. Skyum. On monotone planar circuits. In *IEEE Conference on Computational Complexity (CCC)*, pages 24–31, 1999.

[BT88] G. Di Battista and R. Tamassia. Algorithms for plane representations of acyclic digraphs. *Theoretical Computer Science*, 61:175–198, 1988.

[Bus87] S. Buss. The Boolean formula value problem is in ALOGTIME. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 123–131, 1987.

[CD06] T. Chakraborty and S. Datta. One-input-face MPCVP is hard for L, but in LogDCFL. In *Proceedings of 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 4337 of *Lecture Notes in Computer Science*, pages 57–68, 2006.

[Coo71] S. Cook. Characterizations of pushdown machines in terms of time-bounded computers. *Journal of the Association of Computing Machinery*, 18:4–18, 1971.

[DC89] P. W. Dymond and S. A. Cook. Complexity Theory of Parallel Time and Hardware. *Information and Computation*, 80(3):205–226, 1989.

[DK95] A. L. Delcher and S. R. Kosaraju. An NC algorithm for evaluating monotone planar circuits. *SIAM Journal of Computing*, 24(2):369–375, 1995.

[Fis74] M.J. Fischer. The complexity of negation-limited networks (a brief survey). *Lecture Notes in Computer Science*, 33:71–82, 1974.

[Gol77] L. M. Goldschlager. The monotone and planar circuit value problems are logspace complete for P. *SIGACT News*, 9(2):25–29, 1977.

[Gol80] L. M. Goldschlager. A space efficient algorithm for the monotone planar circuit value problem. *Information Processing Letters*, 10(1):25–27, 1980.

[GT01] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing*, 31(22):601–625, 2001.

[Han06] K. Hansen. Constant width planar computation characterizes ACC^0. *Theory of Computing Systems*, 39(1):79–92, 2006.

[HMV06] K. Hansen, P. Bro Miltersen, and V Vinay. Circuits on cylinders. *Computational Complexity*, 15(1):62–81, May 2006.

[Kel87] D. Kelly. Fundamentals of planar ordered sets. *Discrete Mathematics*, 63(2,3):197–216, 1987.

[Kos90] S. R. Kosaraju. On the parallel evaluation of classes of circuits. In *Proceedings of 10th International Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 472 of *Lecture Notes in Computer Science*, pages 232–237, 1990.

[LMS06] Nutan Limaye, Meena Mahajan, and Jayalal M. N. Sarma. Evaluating monotone circuits on cylinders, planes and tori. In *Proceedings of 23rd International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 3884 of *Lecture Notes in Computer Science*, pages 660–671, 2006.

[Mar58] A. A. Markov. On the inversion complexity of a system of functions. *Journal of the Association of Computing Machinery*, 5(4):331–334, 1958.

[MRK88] G.L. Miller, V. Ramachandran, and E. Kaltofen. Efficient parallel evaluation of straight-line code and arithmetic circuits. *SIAM Journal of Computing*, 17:687–695, 1988.

[MT01] B. Mohar and C. Thomassen. *Graphs on Surfaces*. John Hopkins University Press, Maryland, 2001.

[Rei05] O. Reingold. Undirected *st*-connectivity in logspace. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 376–385, 2005.

[RR94] V. Ramachandran and J. Reif. Planarity testing in parallel. *Journal of Computer and System Sciences*, 49:517–561, 1994.

[Ruz80] W.L. Ruzzo. Tree-size bounded alternation. *Journal of Computer and System Sciences*, 21:218–235, 1980.

[Sud78] I. Sudborough. On the tape complexity of deterministic context-free language. *Journal of the Association of Computing Machinery*, 25(3):405–414, 1978.

[SW91] Miklos Santha and Christopher Wilson. Polynomial size constant depth circuits with a limited number of negations. In *Proceedings of the 8th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 228–237, New York, NY, USA, 1991. Springer-Verlag New York, Inc.

[TT86] R. Tamassia and I. G. Tollis. A unified approach to visibility representations of planar graphs. *Discrete and Computational Geometry*, 1(1):312–341, 1986.

[TT89] R. Tamassia and I. G. Tollis. Tessellation representations of planar graphs. In *Proceedings of the 27th Annual Allerton Conference on Communications, Control and Computing, UIUC*, pages 48–57, 1989.

[Ven91] H. Venkateswaran. Properties that characterize LogCFL. *Journal of Computer and System Sciences*, 42:380–404, 1991.

[Vol99] H. Vollmer. *Introduction to Circuit Complexity: A Uniform Approach.* Springer New York Inc., 1999.

[Whi73] A. T. White. *Graphs, Groups and Surfaces.* North-Holland, Amsterdam, 1973.

[Yan91] H. Yang. An NC algorithm for the general planar monotone circuit value problem. In *Proceedings of the 3rd IEEE Symposium on Parallel and Distributed Processing*, pages 196–203, 1991.

Manuscript received 7 May 2006

Nutan Limaye
The Institute of Mathematical Sciences
Chennai 600 113, India.
nutan@imsc.res.in
http://www.imsc.res.in/~nutan

Meena Mahajan
The Institute of Mathematical Sciences
Chennai 600 113, India.
meena@imsc.res.in
http://www.imsc.res.in/~meena

Jayalal Sarma M.N.
The Institute of Mathematical Sciences
Chennai 600 113, India.
jayalal@imsc.res.in
http://www.imsc.res.in/~jayalal