State Complexity Characterizations of Parameterized Degree-Bounded Graph Connectivity, Sub-Linear Space Computation, and the Linear Space Hypothesis

Tomoyuki Yamakami

State Complexity Characterizations of Parameterized Degree-Bounded Graph Connectivity, Sub-Linear Space Computation, and the Linear Space Hypothesis

Tomoyuki Yamakami

Faculty of Engineering, University of Fukui
3-9-1 Bunkyo, Fukui 910-8507, Japan

Abstract. The linear space hypothesis is a practical working hypothesis, which originally states the insolvability of a restricted 2CNF Boolean formula satisfiability problem parameterized by the number of Boolean variables. From this hypothesis, it follows that the degree-3 directed graph connectivity problem (3DSTCON) parameterized by the number of vertices in a given graph cannot belong to PsubLIN, composed of decision problems computable by polynomial-time, sub-linear-space deterministic Turing machines. This hypothesis immediately implies L≠NL and it was used as a solid foundation to obtain new lower bounds on the computational complexity of various NL search and NL optimization problems. The state complexity of transformation refers to the cost of converting one type of finite automata to another type, where the cost is measured in terms of the increase of the number of inner states of the converted automata from that of the original automata. We relate the linear space hypothesis to the state complexity of transforming restricted 2-way nondeterministic finite automata to computationally equivalent 2-way alternating finite automata having narrow computation graphs. For this purpose, we present state complexity characterizations of 3DSTCON and PsubLIN. We further characterize a non-uniform version of the linear space hypothesis in terms of the state complexity of transformation.

1 Backgrounds and an Overview

1.1 Parameterized Problems and the Linear Space Hypothesis

The *nondeterministic logarithmic-space* complexity class NL has been discussed since early days of computational complexity theory. Typical NL decision problems include the 2CNF Boolean formula satisfiability problem (2SAT) as well as the directed s-t connectivity problem[1] (DSTCON) of determining whether there exists a path from a given vertex s to another vertex t in a given directed graph

[1] This problem is also known as the graph accessibility problem and the graph reachability problem.

G. These problems are also known to be NL-complete under log-space many-one reductions. The NL-completeness is so robust that even if we restrict our interest within graphs whose vertices are limited to be of degree at most 3, the corresponding decision problem, 3DSTCON, remains NL-complete.

In practice, when we measure the computational complexity of given problems, we tend to be more concerned with *parameterizations* of the problems. In other words, we treat the size of specific "input objects" given to the problem as a "practical" *size parameter* n and use it to measure how much resources are needed for algorithms to solve those problems. To emphasize the choice of such a size parameter $m : \Sigma^* \to \mathbb{N}$ for a decision problem L over an alphabet Σ, it is convenient to write (L, m), which gives rise to a *parameterized decision problem.* Since we deal only with such parameterized problems in the rest of this paper, we often drop the adjective "parameterized" as long as it is clear from the context.

Instances $x = \langle G, s, t\rangle$ to 3DSTCON are usually parameterized respectively by the numbers of vertices and of edges in the graph G. It was shown in [2] that DSTCON with n vertices and m edges can be solved in $O(m + n)$ steps using only $n^{1-c/\sqrt{\log n}}$ space for a suitable constant $c > 0$. However, it is unknown whether we can reduce this space usage down to $n^{\varepsilon}\, polylog(m + n)$ for a certain fixed constant $\varepsilon \in [0, 1)$. Such a bound is informally called "sub-linear" in a strong sense. It has been conjectured that, for every constant $\varepsilon \in [0, 1)$, no polynomial-time $O(n^{\varepsilon})$-space algorithm solves DSTCON with n vertices (see references in, e.g., [1, 4]). For convenience, we denote by PsubLIN the collection of all parameterized decision problems (L, m) solvable deterministically in time polynomial in $|x|$ using space at most $m(x)^{\varepsilon}\ell(|x|)$ for certain constants $\varepsilon \in [0, 1)$ and certain polylogarithmic (or polylog, in short) functions ℓ [11].

The *linear space hypothesis* (LSH), proposed in [11], is a practical working hypothesis, which originally asserts the insolvability of a restricted form of 2SAT, denoted $2\mathrm{SAT}_3$, together with the size parameter $m_{vbl}(\phi)$ indicating the number of variables in each given Boolean formula ϕ, in polynomial time using sub-linear space. As noted in [11], it is unlikely that 2SAT replaces $2\mathrm{SAT}_3$. From this hypothesis, nevertheless, we immediately obtain the separation $\mathrm{L} \neq \mathrm{NL}$, which many researchers believe to hold. It was also shown in [11] that $(2\mathrm{SAT}_3, m_{vbl})$ can be replaced by $(\mathrm{3DSTCON}, m_{ver})$, where $m_{ver}(\langle G, s, t\rangle)$ refers to the number of vertices in G. LSH has acted as a reasonable foundation to obtain new lower bounds of several NL-search and NL-optimization problems [11, 12]. To find more applications of this hypothesis, we need to translate the hypothesis into other fields. In this paper, we look for a logically equivalent statement in automata theory, in hope that we would find more applications of LSH in this theory.

1.2 Finite Automata and State Complexity Classes

The purpose of this work is to look for an automata-theoretical statement that is logically equivalent to the linear space hypothesis; in particular, we seek a new characterization of the relationship between 3DSTCON and PsubLIN in terms of the state complexity of transforming a certain type of finite automata to another type with no direct reference to 3DSTCON or PsubLIN.

It is often cited from [3] (re-proven in [8, Section 3]) that, if L = NL, then every n-state two-way nondeterministic finite automaton (or 2nfa) can be converted into an $n^{O(1)}$-state two-way deterministic finite automaton (or 2dfa) that agrees with it on all inputs of length at most $n^{O(1)}$. Conventionally, we call by *unary finite automata* automata working only on unary inputs (i.e, inputs over a one-letter alphabet). Geffert and Pighizzini [6] strengthened the aforementioned result by proving that the assumption of L = NL leads to the following: for any n-state unary 2nfa, there is a unary 2dfa of at most $n^{O(1)}$-states agreeing with it on all strings of length at most n. Within a few years, Kapoutsis [8] gave a similar characterization using L/poly, a non-uniform version of L: NL $\subseteq$ L/poly if and only if (iff) there is a polynomial p such that any n-state 2nfa has a 2dfa of at most $p(n)$ states agreeing with the 2nfa on strings of length at most n. Another incomparable characterization was given by Kapoutsis and Pighizzini [9]: NL $\subseteq$ L/poly iff there is a polynomial p satisfying that any n-state unary 2nfa has an equivalent unary 2dfa of states at most $p(n)$. In this paper, we want to seek a similar automata characterization for the linear space hypothesis.

Sakoda and Sipser [10] further laid out a complexity-theoretical framework to discuss the state complexity by giving formal definitions to state-complexity based classes (such as 2D, 2N/poly, 2N/unary), each of which is generally composed of non-uniform families of languages recognized by finite automata of specified types and input sizes. Such complexity-theoretical treatments of families of finite automata were also considered by Kapoutsis [7, 8] and Kapoutsis and Pighizzini [9]. For those state complexity classes, it was proven in [8, 9] that 2N/poly $\subseteq$ 2D iff NL $\subseteq$ L/poly iff 2N/unary $\subseteq$ 2D.

1.3 Main Contributions

As the main contribution of this paper, firstly we provide with two characterizations of 3DSTCON and PsubLIN in terms of the state complexity of finite automata, and secondly we give a characterization of LSH in terms of the state complexity of transforming a restricted form of 2nfa to another restricted form of two-way alternating finite automaton (or 2afa), which takes $\forall$-states and $\exists$-states alternatingly, producing alternating $\forall$-levels and $\exists$-levels in its (directed) computation graph made up of configurations. The significance of our characterization includes the fact that LSH can be expressed completely by the state complexity of finite automata of certain types *with no clear reference* to $(2\mathrm{SAT}_3, m_{vbl})$, $(3\mathrm{DSTCON}, m_{ver})$, or even PsubLIN; therefore, this characterization may help us apply LSH to a wider range of NL-complete problems.

To describe our result precisely, we further need to explain our terminology. A *simple 2nfa* is a 2nfa having a "circular" input tape[2] (in which both endmarkers are located next to each other) whose tape head "sweeps" the tape (i.e. it moves only to the right), and making nondeterministic choices only at the right endmarker. For a positive integer c, a *c-branching 2nfa* makes only at most c

[2] A 2nfa with a tape head that sweeps a circular tape is called "rotating" in [9].

nondeterministic choices at every step and a family of 2nfa's is called *constant-branching* if there is a constant $c \geq 1$ for which every 2nfa in the family is c-branching. A *c-narrow 2afa* is a 2afa whose computation graphs have width (i.e., the number of distinct vertices at a given level) at every $\forall$-level is bounded by c.

For convenience, we say that a finite automaton M_1 is *equivalent* (in computational power) to another finite automaton M_2 over the same input alphabet if M_1 agrees with M_2 on all inputs. Here, we use a straightforward binary encoding $\langle M \rangle$ of an n-state finite automaton M using $O(n \log n)$ bits. A family $\{M_n\}_{n\in\mathbb{N}}$ is said to be L-*uniform* if a deterministic Turing machine (or a DTM) produces from 1^n an encoding $\langle M_n \rangle$ of finite automaton M_n using space logarithmic in n.

Proposition 1. *Every* L*-uniform family of constant-branching $O(n \log n)$-state simple 2nfa's can be converted into another* L*-uniform family of equivalent $O(n^{1-c/\sqrt{\log n}})$-narrow 2afa's with $n^{O(1)}$ states for a certain constant $c > 0$.*

Theorem 2. *The following three statements are logically equivalent.*
1. *The linear space hypothesis fails.*
2. *For any constants $c > 0$ and $k \geq 1$, there exists a constant $\varepsilon \in [0, 1)$ such that every* L*-uniform family of constant-branching simple 2nfa's with at most $cn \log^k n$ states can be converted into another* L*-uniform family of equivalent $O(n^\varepsilon)$-narrow 2afa's with $n^{O(1)}$ states.*
3. *For any constant $c > 0$, there exists a constant $\varepsilon \in [0, 1)$ and a log-space computable function that, on every input of an encoding of c-branching simple n-state 2nfa, produces another encoding of equivalent $O(n^\varepsilon)$-narrow 2afa with $n^{O(1)}$ states.*

In addition to the original linear space hypothesis, it is possible to discuss its *non-uniform version*, which asserts that $(2\mathrm{SAT}_3, m_{ver})$ does not belong to a non-uniform version of PsubLIN, succinctly denoted by PsubLIN/poly.

The state complexity class 2linN consists of all families $\{L_n\}_{n\in\mathbb{N}}$ of languages, each L_n of which is recognized by a certain c-branching simple $O(n \log^k n)$-state 2nfa on all inputs for appropriate constants $c, k \in \mathbb{N}^+$. Moreover, $2\mathrm{A}_{narrow(f(n))}$ is composed of language families $\{L_n\}_{n\in\mathbb{N}}$ recognized by $O(f(n))$-narrow 2afa's of $n^{O(1)}$ states on all inputs.

Theorem 3. *The following three statements are logically equivalent.*
1. *The non-uniform linear space hypothesis fails.*
2. *For any constant $c > 0$, there exists a constant $\varepsilon \in [0, 1)$ such that every c-branching simple n-state 2nfa can be converted into an equivalent $O(n^\varepsilon)$-narrow 2afa with at most $n^{O(1)}$ states.*
3. $2\mathrm{linN} \subseteq \bigcup_{\varepsilon\in[0,1)} 2\mathrm{A}_{narrow(n^\varepsilon)}$.

It is open whether 2linN in Theorem 3(3) can be replaced by 2N or even 2N/poly. This is related to the question of whether we can replace $2\mathrm{SAT}_3$ in the definition of LSH by 2SAT [11].

In contrast, if we focus our attention on "unary" finite automata, then we obtain a slightly weaker implication to the failure of LSH.

Theorem 4. *Each one of the following statements implies the failure of the linear space hypothesis.*

1. *For any constants $c > 0$ and $k \geq 1$, there exists a constant $\varepsilon \in [0, 1)$ such that every* L*-uniform family of constant-branching simple unary 2nfa's with at most $cn^4 \log^k n$ states can be converted into an* L*-uniform family of equivalent $O(n^\varepsilon)$-narrow unary 2afa's with $n^{O(1)}$ states.*
2. *For any constants $c > 0$ and $k \geq 1$, there exist a constant $\varepsilon \in [0, 1)$ and a log-space computable function that, on every input of an encoding of c-branching simple unary 2nfa with at most $cn^4 \log^k n$ states, produces another encoding of equivalent $O(n^\varepsilon)$-narrow unary 2afa having $n^{O(1)}$ states.*

Theorems 2–3 will be proven in Section 3 after we establish basic properties of PsubLIN and 3DSTCON in Section 2. Theorem 4 will be shown in Section 4.

2 Two Basic Characterizations

Since Theorems 2–3 are concerned with 3DSTCON and PsubLIN, we want to look into their basic properties. In what follows, we will present two state complexity characterizations of the complexity class PsubLIN and the language 3DSTCON.

A function $m : \Sigma^* \to \mathbb{N}^+$ is called a *log-space size parameter* if there exists a DTM M that, on any input x, produces $m(x)$ in binary on its output tape using only $O(\log |x|)$ work space. A function $f : \mathbb{N} \to \mathbb{N}$ is *log-space computable* (resp., $t(n)$*-time space constructible* for a given function t) if there exists a DTM with a write-only output tape such that, for each given length $n \in \mathbb{N}$, when M takes 1^n and then produces $1^{f(n)}$ using $O(\log n)$ work space (resp., within $O(t(n))$ steps using no more than $f(n)$ cells).

2.1 Automata Characterizations of PsubLIN

Let us give a precise characterization of PsubLIN in terms of the state complexity of narrow 2afa's because the narrowness of 2afa's directly corresponds to the space usage of DTMs. What we intend to prove in this section is, in fact, slightly more general than what we actually need for proving Theorems 2–3.

Take two functions $s : \mathbb{N} \times \mathbb{N} \to \mathbb{N}^+$ and $t : \mathbb{N} \to \mathbb{N}^+$, and let m denote any log-space size parameter, where $\mathbb{N}^+ = \mathbb{N} - \{0\}$. We define $\mathrm{TIME}, \mathrm{SPACE}(t(x), s(x, m(x)))$ (where x expresses a symbolic input) to be the collection of all parameterized decision problems (L, m) recognized by DTMs (each of which is equipped with a read-only input tape and a semi-infinite rewritable work tape) within time $c_1 t(x)$ using space at most $c_2 s(x, m(x))$ on every input x for certain absolute constants $c_1, c_2 > 0$.

Our proof of Proposition 5 is a fine-grained analysis of the well-known transformation of *alternating Turing machines* (or ATMs) to DTMs and vice versa. In what follows, we freely identify a language with its *characteristic function.*

Proposition 5. *Let $t, \ell : \mathbb{N} \to \mathbb{N}^+$ be log-space computable and $O(t(n))$-time space constructible, respectively. Consider a language L and a log-space size parameter m.*

1. *If* $(L,m) \in \mathrm{TIME}, \mathrm{SPACE}(t(|x|), \ell(m(x)))$*, then there are two constants* $c_1, c_2 > 0$ *and an* L*-uniform family* $\{M_{n,l}\}_{n,l\in\mathbb{N}}$ *of* $c_2\ell(m(x))$*-narrow 2afa's such that each* $M_{n,|x|}$ *has at most* $c_1 t(|x|)\ell(m(x))$ *states and computes* $L(x)$ *on all inputs* x *satisfying* $m(x) = n$*.*

2. *If there are constants* $c_1, c_2 > 0$ *and an* L*-uniform family* $\{M_{n,l}\}_{n,l\in\mathbb{N}}$ *of* $c_2\ell(m(x))$*-narrow 2afa's such that each* $M_{n,|x|}$ *has at most* $c_1 t(|x|)$ *states and computes* $L(x)$ *on all inputs* x *satisfying* $m(x) = n$*, then* (L,m) *belongs to* $\mathrm{TIME}, \mathrm{SPACE}(t(|x|)\ell(m(x)), \ell(m(x)) + \log t(|x|) + \log |x|)$*.*

Proof Sketch. (1) Given a parameterized decision problem (L,m), let us consider a DTM N that solves (L,m) in time at most $c_1 t(|x|)$ using space at most $c_2\ell(m(x))$ for certain constants $c_1, c_2 > 0$. We first modify N so that it halts in scanning both ¢ on the input tape and the blank symbol B at the *start cell* (i.e., cell 0) of the work tape. Moreover, we make it halt in exactly $c_1 t(|x|)$ steps. Now, we want to simulate N by 2afa's $M_{n,l}$ of the desired type. Let x be any instance to L. Let us consider *surface configurations* (q,j,k,w) of N on x, which indicates that N is in state q, scanning both the jth cell of the input tape and the kth cell of the work tape containing w. We want to trace down these surface configurations using an alternating series of $\forall$-states and $\exists$-states of $M_{n,|x|}$.

Since each move of N affects at most 3 consecutive cells of the input tape and the work tape, it suffices to focus our attention on these local cells. Our idea is to define $M_{n,|x|}$'s surface configuration $((q,i,k',u),j)$ to represent N's surface configuration (q,j,k,w) at time i in such a way that u indicates either the k'-th cell content or the content of its neighboring 3 cells. In particular, when $k = k'$, u carries extra information (by changing tape symbol σ to $\hat{\sigma}$) that a tape head is at the k'th cell. For example, an initial surface configuration of $M_{n,|x|}$ on x is $((q_{acc}, c_1 t(|x|), 0, \hat{B}), 0)$, which corresponds to the final accepting surface configuration of N on x, where q_{acc} is a unique accepting state of N. Inductively, we generate the next surface configuration of $M_{n,|x|}$ roughly in the following way. In an $\exists$-state, $M_{n,|x|}$ guesses (i.e., nondeterministically chooses) the content of 3 consecutive cells in the current configuration of N on x. In a $\forall$-state, $M_{n,|x|}$ checks whether the guessed content is actually correct by branching out 3 computation paths, each of which selects one of the 3 cells chosen in the $\exists$-state. The $O(\ell(m(x))$-narrowness comes from the space bound of N.

(2) Let $k \geq 1$ and $\mathcal{M} = \{M_{n,l}\}_{n,l\in\mathbb{N}}$ be a family given for L by the premise of (2). In particular, each $M_{n,l}$ is a $c_2\ell(m(x))$-narrow 2afa having at most $c_1 t(|x|)$ states for constants $c_1, c_2 > 0$. We simulate $\mathcal{M}$ by the following DTM. On input x, compute $n = m(x)$, and generate $\langle M_{n,|x|}\rangle$ using $O(\log |x|)$ space. Consider a computation graph of $M_{n,|x|}$ on input x. Using a breadth-first search technique, we check whether there is an accepting computation subgraph of $M_{n,|x|}$ on x by trimming all encountered branches that lead to rejecting states. It is possible to carry out this procedure using space $O(\log t(|x|)) + O(\ell(m(x))) + O(\log |x|)$

since $M_{n,|x|}$ is $c_2\ell(m(x))$-narrow and $O(\log t(|x|))$ bits are needed to describe each state. The running time of this DTM is at most $O(t(|x|)\ell(m(x)))$. □

Similarly, we can obtain a non-uniform version of Proposition 5 by making use of "advice" instead of the uniformity condition. In the uniform case, we have used a DTM to produce $\langle M_n \rangle$ from 1^n; in the non-uniform case, by contrast, we must generate $\langle M_n \rangle$ from information given by the advice.

A Karp-Lipton non-uniform version of $\mathrm{TIME,SPACE}(t(x), \ell(x, m(x)))$, which is denoted by $\mathrm{TIME,SPACE}(t(x), \ell(x, m(x)))/O(s(|x|))$, is defined by supplementing external information known as "advice" to underlying Turing machines that characterize $\mathrm{TIME,SPACE}(t(x), \ell(x, m(x)))$. Each of such underlying machines is equipped with an additional read-only *advice tape*, to which we provide exactly one string (called an *advice string*) of length $O(s(n))$ surrounded by two endmarkers for all input instances of length n.

Proposition 6. *Let $t : \mathbb{N} \to \mathbb{N}^+$ be log-space computable and let $s, \ell : \mathbb{N} \to \mathbb{N}^+$ be $O(t(n))$-time space constructible. Let L and m be a language and a log-space size parameter, respectively. Assume that there is a function h satisfying $|x| \leq h(m(x))$ for all x.*

1. *If $(L, m) \in \mathrm{TIME,SPACE}(t(|x|), \ell(m(x))/O(s(|x|))$, then there is a non-uniform family $\{M_{n,l}\}_{n,l\in\mathbb{N}}$ of $O(\ell(m(x)))$-narrow 2afa's such that, for each $n \in \mathbb{N}$, $M_{n,|x|}$ has $O(t(|x|)\ell(m(x))s(|x|))$ states and computes $L(x)$ on all inputs x satisfying $m(x) = n$.*
2. *If there is a non-uniform family $\{M_{n,l}\}_{n,l\in\mathbb{N}}$ of $O(\ell(m(x)))$-narrow 2afa's such that each $M_{n,|x|}$ has $O(t(|x|))$ states and computes $L(x)$ on all inputs x satisfying $m(x) = n$, then (L, m) belongs to $\mathrm{TIME,SPACE}(t(|x|)\ell(m(x)), \ell(m(x)) + \log t(|x|) + \log |x|)/O(h(m(x))t(|x|)^2 \log t(|x|))$.*

2.2 Automata Characterizations of 3DSTCON

The proofs of Theorems 2 and 3 require a characterization of 3DSTCON in terms of 2nfa's. Kapoutsis [8] and Kapoutsis and Pighizzini [9] earlier gave 2nfa-characterizations of DSTCON; however, 3DSTCON needs a slightly different characterization.

First, we re-formulate the parameterized decision problem $(\mathrm{3DSTCON}, m_{ver})$ as a family $\{\mathrm{3DSTCON}_n\}_{n\in\mathbb{N}}$ of decision problems, each $\mathrm{3DSTCON}_n$ of which is limited to directed graphs of vertex size exactly n. To express instances to $\mathrm{3DSTCON}_n$, we also need to define an appropriate binary encoding of degree-bounded directed graphs. Formally, let $K_n = (V, E)$ denote a complete directed graph with $V = \{0, 1, \ldots, n-1\}$ and $E = V \times V$ and let $G = (V, E)$ be a degree-3 subgraph of K_n. We express this graph G as the form of an *adjacency list*, which is represented by an $n \times 3$ matrix whose rows are indexed by $i \in [n]$ and columns are indexed by $j \in \{1, 2, 3\}$. If there is no jth edge leaving from vertex i, then the (i, j)th entry of this list is the designated symbol $\perp$. We further encode this list into a single binary string, denoted by $\langle G \rangle$, of size $O(n \log n)$. Here, we demand that we can easily check whether a given string is an binary encoding of a certain directed graph.

Lemma 7. *There exists an* L*-uniform family* $\{N_n\}_{n\in\mathbb{N}}$ *of* $O(n\log n)$*-state simple 2dfa's, each* N_n *of which checks whether any given input* x *is an encoding* $\langle G\rangle$ *of a certain subgraph* G *of* K_n.

The language family $\{\text{3DSTCON}_n\}_{n\in\mathbb{N}}$ is defined as follows.

Degree-3 Directed s-t Connectivity Problem for Size n (3DSTCON_n):

- ◦ Instance: an encoding $\langle G\rangle$ of a subgraph G of the complete directed graph K_n with vertices of degree (i.e., indegree plus outdegree) at most 3.
- ◦ Output: YES if there is a path from vertex 0 to vertex $n-1$; NO otherwise.

Notice that each instance x belonging to 3DSTCON_n must satisfy $m_{ver}(x) = n$. Clearly, the family $\{\text{3DSTCON}_n\}_{n\in\mathbb{N}}$ corresponds to $(\text{3DSTCON}, m_{ver})$, and thus we freely identify $(\text{3DSTCON}, m_{ver})$ with the family $\{\text{3DSTCON}_n\}_{n\in\mathbb{N}}$.

Lemma 8. *There is an absolute constant* $c > 0$ *such that* $m_{ver}(x) \leq |x| \leq cm_{ver}(x)\log m_{ver}(x)$ *for all inputs* x *to* 3DSTCON.

Next, we want to build a uniform family of constant-branching simple 2nfa's that solve $\{\text{3DSTCON}_n\}_{n\in\mathbb{N}}$. Let Σ_n denote the set of all valid encodings of input graphs given to 3DSTCON_n.

Lemma 9. *There exists an* $O(\log n)$*-space computable function* g *for which* g *produces from each* 1^n *a description of* 3*-branching simple 2nfa* N_n *of* $O(n\log n)$ *states that solves* 3DSTCON_n *on inputs in* Σ_n *in time* $O(n^2)$*. Moreover,* N_n *can reject all inputs outside of* Σ_n.

Proof Sketch. Our 2nfa has a circular tape and moves its tape head only to the right. Choose any input $x = \langle G\rangle$ to 3DSTCON_n, where $G = (V, E)$ is a degree-3 subgraph of K_n with $V = \{0, 1, \ldots, n-1\}$.

We design M_n so that it works *round by round* in the following way. At the first round, we assign vertex 0 in G to v_0 and move the tape head rightward from ¢ to \$. Now, assume by induction hypothesis that, at round i (≥ 0), we have already chosen vertex v_i and have moved the tape head to \$. Nondeterministically, we select an index $j \in \{1, 2, 3\}$ while scanning \$, and then deterministically search for a row indexed i in an adjacency list of G by moving the tape head only from left to right along the circular tape. We then read the content of the (i, j)-entry of the list. If it is $\perp$, then reject immediately. Assuming otherwise, if v_{i+1} is the (i, j)-entry, then we update the current vertex from v_i to v_{i+1}. Whenever we reach vertex $n-1$, we immediately accept x and halt. If M_n visits more than n vertices, we surely know that M_n cannot accept x. □

Let us consider the converse of Lemma 9.

Lemma 10. *Let* $c \in \mathbb{N}^+$ *be a constant. There exists a function* g *such that, for every* c*-branching simple 2nfa* M *with* n *states,* g *takes input* $\langle M\rangle\#x$ *and outputs an encoding* $\langle G_x\rangle$ *of a subgraph* G_x *of* K_{2n+3} *of degree at most* $2(c+1)$ *satisfying that* M *accepts* x *if and only if* $G_x \in \text{3DSTCON}_{2n+3}$*. Moreover,* g *is computed by a certain* $n^{O(1)}$*-state simple 2dfa with a write-only output tape.*

3 Proofs of Theorems 2 and 3

3.1 Generalizations to PTIME,SPACE(·)

Theorems 2 and 3(1)–(2) are concerned with PsubLIN. Nonetheless, it is possible to prove slightly more general theorems, shown below as Theorems 11 and 12, for a complexity class $\mathrm{PTIME,SPACE}(s(x,m(x)))$, defined in [11], which is the union of all $\mathrm{TIME,SPACE}(p(|x|),s(x,m(x)))$ for any positive polynomial p.

Theorem 11. *Let $\mathcal{F}$ denote an arbitrary nonempty set of functions $\ell:\mathbb{N}\to\mathbb{N}^+$ such that, for every $\ell\in\mathcal{F}$ and every $c>0$ and $k\in\mathbb{N}^+$, there are functions $\ell',\ell''\in\mathcal{F}$ such that $\ell(cn\log^k n)\leq\ell'(n)$ and $\ell(n)+\log n^k\leq\ell''(n)$ for all $n\in\mathbb{N}$. Assume that, for each $\ell\in\mathcal{F}$, $\bigcup_m \mathrm{PTIME,SPACE}(\ell(m(x)))$ is closed under short L-m-reductions (see [11]), where m ranges over all log-space size parameters. The following three statements are logically equivalent.*

1. *There exists a function $\ell\in\mathcal{F}$ such that $(\mathrm{3DSTCON},m_{ver})$ is in $\bigcup_m \mathrm{PTIME,SPACE}(\ell(m(x)))$.*
2. *There are a function $\ell\in\mathcal{F}$ and two constants $c>0$ and $k\in\mathbb{N}^+$ such that every L-uniform family of constant-branching simple 2nfa's with at most $cn\log^k n$ states is converted into another L-family of $O(\ell(n))$-narrow 2afa's with $n^{O(1)}$ states that agree with them on all inputs.*
3. *There are a function $\ell\in\mathcal{F}$ and a constant $\varepsilon\in[0,1)$ satisfying the following: for each constant $c\in\mathbb{N}^+$, there exists a log-space computable function f such that f takes an input of the form $\langle M\rangle$ for any c-branching n-state simple 2nfa M and f produces another encoding of $O(\ell(n))$-narrow 2afa with $n^{O(1)}$ states that agree with M on all inputs.*

Theorem 12. *Let $\mathcal{F}$ denote an arbitrary nonempty set of functions $\ell:\mathbb{N}\to\mathbb{N}^+$ such that, for every $\ell\in\mathcal{F}$ and every $c>0$ and $k\in\mathbb{N}^+$, there are functions $\ell',\ell''\in\mathcal{F}$ such that $\ell(cn\log^k n)\leq\ell'(n)$ and $\ell(n)+\log n^k\leq\ell''(n)$ for all $n\in\mathbb{N}$. Assume that $\bigcup_m \mathrm{PTIME,SPACE}(\ell(m(x)))/\mathrm{poly}$ is closed under short L-m-reductions, where m is any log-space size parameter. There is an $\ell\in\mathcal{F}$ such that $(\mathrm{3DSTCON},m_{ver})$ is in $\bigcup_m \mathrm{PTIME,SPACE}(\ell(m(x)))/\mathrm{poly}$ iff, for each constant $e\in\mathbb{N}^+$, there are an $\ell\in\mathcal{F}$ and a constant $\varepsilon\in[0,1)$ such that any n-state e-branching simple 2nfa can be converted into another $n^{O(1)}$-state $O(\ell(n))$-narrow 2afa that agrees with it on all inputs.*

Proof of Theorems 2 and 3(1)–(2). These results follow from the fact that Theorems 2 and 3(1)–(2) are special cases of Theorems 11 and 12, respectively, where $\ell(n)$ equals n^{ε} for a certain constant $\varepsilon\in[0,1)$. □

Now, we return to Theorem 11 and describe its proof.

Proof Sketch of Theorem 11. For convenience, given a function ℓ, we write $\mathcal{C}_\ell$ for $\bigcup_m \mathrm{PTIME,SPACE}(\ell(m(x)))$ regarding all log-space size parameters m.

[1 ⇒ 3] Assume that $(\mathrm{3DSTCON},m_{ver})\in\mathcal{C}_\ell$ for a certain function $\ell\in\mathcal{F}$. Let $c>0$ be a constant. By applying Proposition 5(1), we obtain a constant

$k \geq 1$ and an L-uniform family $\{N_{n,l}\}_{n,l\in\mathbb{N}}$ of $O(\ell(m_{ver}(x)))$-narrow 2afa's having $O(|x|^k \cdot m_{ver}(x))$ states that agree with 3DSTCON(x) on all inputs x satisfying $m_{ver}(x) = n$. Take a log-space computable function g that produces $\langle N_{n,l}\rangle$ from input $1^n\#1^l$. For simplicity, let $d = 2n+3$. By Lemma 10, there is a function g that transforms $\langle M\rangle\#x$ to the encoding $\langle G_x\rangle$ of a subgraph G_x of K_n satisfying that M accepts x exactly when $G_x \in \mathrm{3DSTCON}_d$. Note that g is computed by a certain simple 2dfa with $n^{O(1)}$ states.

Next, we want to design a log-space computable function f that transforms every c-branching n-state simple 2nfa M to another 2afa N of the desired type. We define f so that, given an encoding $\langle M\rangle$ of a c-branching simple 2nfa M with n states, it produces an appropriate 2afa N that works as follows. On input x, generate $\langle G_x\rangle$ from $\langle M\rangle\#x$ by applying g and compute $e = |\langle G_x\rangle|$, which is $O(n\log n)$. Produce $N_{d,e}$ and run it on the input $\langle G_x\rangle$. Note that we cannot actually write down $\langle G_x\rangle$ onto a tape; however, since g is computed by a simple 2dfa, we can produce every bit of $\langle G_x\rangle$ separately.

[3 $\Rightarrow$ 2] Assuming (3), we obtain a log-space computable function g that, from any c-branching n-state simple 2nfa, produces an $\ell(n)$-narrow 2afa that agrees with it on all inputs. Let us take any L-uniform family $\{M_n\}_{n\in\mathbb{N}}$ of simple 2nfa's, each M_n of which has at most $cn\log^k n$ states for absolute constants $c > 0$ and $k \in \mathbb{N}^+$. By the L-uniformity of $\{M_n\}_{n\in\mathbb{N}}$, we choose a log-space DTM N that produces $\langle M_n\rangle$ from 1^n for each $n \in \mathbb{N}$. By (3), we obtain an $\ell(cn\log^k n)$-narrow 2afa $\langle N_n\rangle$ from $\langle M_n\rangle$ in polynomial time using log space. Hence, $\{N_n\}_{n\in\mathbb{N}}$ is L-uniform. Moreover, by our assumption, there is a function $\ell' \in \mathcal{F}$ such that $\ell(cn\log^k n) \leq \ell'(n)$ for all $n \in \mathbb{N}$. It thus follows that N_n is $\ell'(n)$-narrow.

[2 $\Rightarrow$ 1] Let $c \geq 1$. Assume that we can convert any L-uniform family of c-branching $cn\log^k n$-state simple 2nfa's into another L-uniform family of $n^{O(1)}$-state $O(\ell(n))$-narrow 2afa that agrees with it on all inputs, where $\ell \in \mathcal{F}$. Let us consider $\{\mathrm{3DSTCON}_n\}_{n\in\mathbb{N}}$. By Lemma 9, we obtain an an L-uniform family of 3-branching simple 2nfa's N_n with $cn\log n$ states that solve $\mathrm{3DSTCON}_n$ within cn^2 steps on all inputs x with $m_{ver}(x) = n$, where $c \geq 3$ is an appropriate constant.

Since $m_{ver}(x) \leq |x| \leq em_{ver}(x)\log m_{ver}(x)$ for a certain constant $e > 0$ by Lemma 8, we obtain $|x| \leq en\log n$. Apply (2), and we obtain 2afa's, which has $n^{O(1)}$ states and is $O(\ell(cn\log n))$-narrow, solving $\{\mathrm{3DSTCON}_n\}_{n\in\mathbb{N}}$ on all inputs, including all strings x satisfying $m_{ver}(x) = n$. Take an $\ell' \in \mathcal{F}$ such that $\ell'(n) \geq \ell(cn\log n)$ for all $n \in \mathbb{N}$. By Proposition 5(2), we conclude that $(\mathrm{3DSTCON}, m_{ver})$ belongs to $\mathrm{TIME,SPACE}(|x|^{O(1)}, \ell'(m_{ver}(x)))$, which is included in $\mathcal{C}_{\ell'}$. □

The proof of Theorem 12 is in essence similar to that of Theorem 11 except for the treatment of advice strings.

An argument similar to that of [1 $\Rightarrow$ 3] in the proof of Theorem 11 leads to Proposition 1 on top of the result of Barnes et al. [2] on $(\mathrm{DSTCON}, m_{ver})$.

3.2 Relationships among State Complexity Classes

To complete the proof of Theorem 3, nevertheless, we still need to show the logical equivalence between (1) and (3) of the theorem. To achieve this goal, we first show Proposition 13, in which we present a close relationship between PsubLIN and $\bigcup_{\varepsilon\in[0,1)} 2\mathrm{A}_{narrow(n^\varepsilon)}$.

Proposition 13. *Given a parameterized decision problem (L, m) with a log-space size parameter m, define $L_{n,l} = \{x \in L \cap \Sigma^l \mid m(x) = n\}$ and $\overline{L}_{n,l} = \{x \in \overline{L} \cap \Sigma^l \mid m(x) = n\}$ for each pair $n, l \in \mathbb{N}$. We set $\mathcal{L} = \{(L_{n,l}, \overline{L}_{n,l})\}_{n,l\in\mathbb{N}}$. Assume that there are constants $c_1, c_2 > 0$ and $k \geq 1$ for which $c_1 m(x) \leq |x| \leq c_2 m(x) \log^k m(x)$ for all x with $|x| \geq 2$. It then follows that $(L, m) \in$ PsubLIN/poly iff $\mathcal{L} \in \bigcup_{\varepsilon\in[0,1)} 2\mathrm{A}_{narrow(n^\varepsilon)}$.*

Corollary 14. $(\mathrm{3DSTCON}, m_{ver}) \in$ PsubLIN/poly *if and only if* $\{\mathrm{3DSTCON}_n\}_{n\in\mathbb{N}} \in \bigcup_{\varepsilon\in[0,1)} 2\mathrm{A}_{narrow(n^\varepsilon)}$.

Proof Sketch of Theorem 3(3). Write $\mathcal{L}$ for $\{\mathrm{3DSTCON}_n\}_{n\in\mathbb{N}}$. By Lemma 9, we obtain $\mathcal{L} \in$ 2linN. It is possible to prove that (*) 2linN $\subseteq \bigcup_{\varepsilon\in[0,1)} 2\mathrm{A}_{narrow(n^\varepsilon)}$ iff $\mathcal{L} \in \bigcup_{\varepsilon\in[0,1)} 2\mathrm{A}_{narrow(n^\varepsilon)}$. The equivalence between (1) and (3) of Theorem 3 follows directly from Corollary 14 and the above statement (*). □

4 Case of Unary Finite Automata

The proof of Theorem 4 needs a unary version of $\{\mathrm{3DSTCON}_n\}_{n\in\mathbb{N}}$. Hence, we first define a *unary encoding* of a degree-bounded subgraph of each complete directed graph K_n. Given a degree-3 subgraph $G = (V, E)$ of K_n with $V = \{0, 1, 2, \ldots, n-1\}$, the unary encoding $\langle G\rangle_{unary}$ of G is of the form 1^e with $e = \prod_{l=1}^{k} p_{(i_l,j_l)}$, where $E = \{(i_1, j_1), (i_2, j_2), \ldots, (i_k, j_k)\} \subseteq V^2$ with $k = |E|$ and each $p_{(i,j)}$ denotes the $(i \cdot n + j)$-th prime number. Since G has degree at most 3, it follows that $k \leq 3n$. It is known that the rth prime number is at most $cr \log r$ for a certain constant $c > 0$. Since $i \cdot n + j \leq n^2$ for all pairs $i, j \in V$, we conclude that $|\langle G\rangle_{unary}| = e \leq (cn^2 \log n)^{3n}$. Let $\{\mathrm{unary3DSTCON}_n\}_{n\in\mathbb{N}}$ be defined as follows.

Unary 3DSTCON of Size n ($\mathrm{unary3DSTCON}_n$):

- Instance: $\langle G\rangle_{unary}$ for a subgraph G of K_n with vertices of degree at most 3.
- Output: YES if there is a path from vertex 0 to vertex $n-1$; NO otherwise.

Proof Sketch of Theorem 4. (1) Assume that every L-uniform family of $O(n^4 \log^k n)$-state constant-branching simple unary 2nfa's can be converted into another L-uniform family of equivalent $n^{O(1)}$-state $O(n^\varepsilon)$-narrow unary 2afa's. We then take a function g that transforms $\langle G\rangle$ to $\langle G\rangle_{unary}$. Note that g can be implemented by an appropriate L-uniform family of $n^{O(1)}$-state simple 2dfa's. We further take a constant $c > 0$ and an L-uniform family $\{M_n\}_{n\in\mathbb{N}}$ of c-branching simple 2nfa's of $O(n^4 \log n)$ states, each M_n of which solves unary3DSTCON$_n$ on all inputs. Our assumption guarantees the existence of an L-uniform family $\{N_n\}_{n\in\mathbb{N}}$ of $O(n^\varepsilon)$-narrow 2afa's with $n^{O(1)}$ states, each N_n of which agrees with M_n on all inputs for a suitable choice of constant $\varepsilon \in [0,1)$.

We want to show that the condition of Theorem 2(2) is satisfied. Let $\langle G\rangle$ be any input to 3DSTCON$_n$. To this input $\langle G\rangle$, we apply g in order to produce $\langle G\rangle_{unary}$ and then run N_n on $\langle G\rangle_{unary}$. This new 2afa has $n^{O(1)}$ states and is $O(n^\varepsilon)$-narrow, as we expected. Thus, the desired condition holds.

(2) Assume that, given a $k \in \mathbb{N}^+$, there are a constant $\varepsilon \in [0,1)$ and a log-space computable function g for which, on each input $\langle M\rangle$ of c-branching simple 2nfa M with $O(n^4 \log^k n)$ states, g outputs its equivalent $O(n^\varepsilon)$-narrow 2afa N. It suffices to consider the following machine. On input $\langle G\rangle$ to 3DSTCON$_n$, we transform it to $\langle G\rangle_{unary}$ and run M_{2n+3} on $\langle G\rangle_{unary}$. This makes the condition of Theorem 2(3) true. □

References

1. Allender, E., Chen, S., Lou, T., Papakonstantinou, P.A., Tang, B.: Width-parameterized SAT: time-space tradeoffs. Theory of Computing 10 (2014) 297–339.
2. Barnes, G., Buss, J.F., Ruzzo, W.L., Schieber, B.: A sublinear space, polynomial time algorithm for directed s-t connectivity. SIAM J. Comput. 27, 1273–1282 (1998)
3. Berman, P., Lingas, A.: On complexity of regular languages in terms of finite automata. Report 304, Institute of Computer Science, Polish Academy of Science, Warsaw, (1977)
4. Chakraborty, D., Tewari, R.: Simulteneous time-space upper bounds for red-blue path problem in planar DAGs. WALCOM 2015, LNCS, vol. 8973, pp. 258–269 (2015)
5. Geffert, V., Mereghetti, C., Pighizzini, G.: Converting two-way nondeterministic automata into simpler automata. Theoret. Comput. Sci. 295, 189–203 (2003)
6. Geffert, V., Pighizzini, G.: Two-way unary automata versus logarithmic space. Inform. Comput. 209, 1016–1025 (2011)
7. Kapoutsis, C.A.: Minicomplexity. J. Automat. Lang. Combin. 17, 205-224 (2012)
8. Kapoutsis, C.A.: Two-way automata versus logarithmic space. Theory Comput. Syst. 55, 421–447 (2014)
9. Kapoutsis, C.A., Pighizzini, G.: Two-way automata characterizations of L/poly versus NL. Theory Comput. Syst. 56, 662–685 (2015)
10. Sakoda, W.J., Sipser, M.: Nondeterminism and the size of two-way finite automata. STOC 1978, pp. 275–286 (1978)
11. Yamakami, T.: The 2CNF Boolean formula satsifiability problem and the linear space hypothesis. MFCS 2017, LIPIcs vol. 83, 62:1–62:14 (2017). Available also at arXiv:1709.10453.
12. Yamakami, T.: Parameterized graph connectivity and polynomial-time sub-linear-space short reductions (preliminary report). RP 2017, LNCS, vol. 10506, pp. 176–191 (2017)