International Journal of Foundations of Computer Science

ON THE ISOMORPHISM CONJECTURE FOR 2-DFA REDUCTIONS

MANINDRA AGRAWAL
Department of Computer Science and Engineering, Indian Institute of Technology
Kanpur 208016, India

and

S. VENKATESH
Computer Science Group, Tata Institute of Fundamental Research, Homi Bhabha Road, Colaba
Bombay 400005, India

Received 4 January 1995
Revised October 3, 2006
Communicated by J. Y. Cai

ABSTRACT

The degree structure of complete sets under 2DFA reductions is investigated. It is shown that, for any class $\mathcal{C}$ that is closed under log-lin reductions:

- All complete sets for the class $\mathcal{C}$ under 2DFA reductions are also complete under one-one, length-increasing 2DFA reductions and are first-order isomorphic.
- The 2DFA-isomorphism conjecture is false, i.e., the complete sets under 2DFA reductions are not isomorphic to each other via 2DFA reductions.

Keywords: Isomorphism; Completeness; Reducibility; Computational Complexity.

1. Introduction

The notion of NP-completeness has been central to the development of complexity theory. Berman and Hartmanis[1]—motivated by analogous results in recursion theory[2] and the fact that all known NP-complete sets at the time were p-isomorphic to each other—conjectured that all NP-complete sets are p-isomorphic to each other. That is, all NP-complete sets are essentially the same except for polynomial-time reencodings. This conjecture has been a target of much research over the years (see Ref. [3] for a survey) but no definite answer has been given yet—though there are some indications that it is false.[4,5] Attempts have also been made to answer the conjecture for classes other than NP and reducibilities other than polynomial-time. For a class $\mathcal{C}$ and reducibility r, the *r-isomorphism conjecture for $\mathcal{C}$* is that all $\leq_m^r$-complete sets for $\mathcal{C}$ are r-isomorphic to each other (here the bijection is an r-computable function in both directions). While the answer of the p-isomorphism conjecture for classes other than NP also remains elusive, for some reducibilities that are weaker than polynomial-time the conjecture has been settled.[6] In particular, it has been shown that for 1-NL functions—functions computed by nondeterministic logspace TMs with a one-way input head—the 1-NL-isomorphism conjecture is true for any class closed under log-lin reductions (see next section for the definition of log-lin functions). On the other hand, for 1-L reductions—functions computed by deterministic logspace TMs with a one-way input head—the 1-L-isomorphism conjecture is false for any class closed under log-lin reductions.

In this paper, we investigate the isomorphism conjecture for *2DFA reductions*. These reductions are computed by DTMs with a read-only input tape, a write-only output tape, and no workspace. They were defined in Ref. [7] where several problems complete for E under these reductions were exhibited. The complete degrees of these reductions have been investigated earlier too.[8,9,10] In Ref. [8] it was shown that for DSPACE(n) and every reasonable deterministic class above it, $\leq_m^{2dfa}$-complete sets are p-isomorphic; in Ref. [9] the same result was shown for every reasonable nondeterministic class above and including NE; and in Ref. [10] it was shown that for every class $\mathcal{C}$ closed under log-lin reductions, $\leq_m^{2dfa}$-complete sets for $\mathcal{C}$ are p-isomorphic.

We improve upon all the above result—we show that for any class $\mathcal{C}$ that is closed under log-lin reductions, all $\leq_m^{2dfa}$-complete sets for $\mathcal{C}$ are also complete under one-one and length-increasing 2DFA reductions and are isomorphic under *first-order* reductions.[11] However, the 2DFA isomorphism conjecture fails to hold—we show that for every class closed under log-lin reductions, the 2DFA-isomorphism conjecture is false.

The paper is organized as follows. In the next section, we introduce the basic definitions. In section three, we define the notion of forgetful 2DFA TMs that plays a key role in our collapse result. In section four, we prove our results for the complete degrees under 2DFA reductions. We end with some discussion on the results obtained in section five.

2. Preliminaries

We assume that the reader is familiar with the basic notion of Turing Machines and complexity classes.[12]

The strings are over $\{0,1\}$. For any string $w \in \{0,1\}^*$, $|w|$ will denote its length and for any number $i \leq |w|$, we denote the i^{th} bit of w as $w[i]$.

For a resource bound r on TMs, we denote by $\mathcal{F}(r)$ the class of total functions computed by TMs within the resource bound of r. For the class of functions $\mathcal{F}(r)$, we say that f is an *r-computable function*, or simply, an *r function*, if $f \in \mathcal{F}(r)$; and f is *r-invertible* if there is a function $g \in \mathcal{F}(r)$ such that $g(f(x)) = x$ for every x. We say that a set $A \leq_m^r$ ($\leq_{1,li}^r$; $\leq_{1,li,i}^r$) B if there is a many-one (one-one, length-increasing; one-one, length-increasing and r-invertible) r-computable function f reducing A to B. Set A is *$\leq_m^r$-hard* for class $\mathcal{C}$ if for every $B \in \mathcal{C}$, $B \leq_m^r A$. Set A is *$\leq_m^r$-complete* for class $\mathcal{C}$ if A is $\leq_m^r$-hard for $\mathcal{C}$ and $A \in \mathcal{C}$. For the class NP, an NP-*complete* set is a $\leq_m^p$-complete set for NP. The *$\leq_m^r$-complete degree* of $\mathcal{C}$ is defined to be the class of all $\leq_m^r$-complete sets for $\mathcal{C}$. Similarly, one defines these notions for $\leq_{1,li}^r$, and $\leq_{1,li,i}^r$ reductions. We say that the set A is *r-isomorphic* to the set B if there exists a bijection f between A and B with both f and f^{-1} being r-computable.

A *2DFA* TM is a deterministic TM with a read-only input tape, a write-only output tape and a two-way input head. It is not clear if the class of total 2DFA functions, $\mathcal{F}$(2DFA), is closed under composition. Nevertheless, the $\leq_m^{2dfa}$-complete degree is well defined for any class.

Finally, function f is a *log-lin* function if there is a logspace bounded DTM that computes f and for every x, $|f(x)| = O(|x|)$.[7]

3. Forgetful 2DFA TMs

In this section, we introduce the notion of forgetful 2DFA TMs and prove a key theorem that will be at the heart of the proof of our result about the collapse of $\leq_m^{2dfa}$-complete degrees.

For any 2DFA TM, we can assume the following without loss of generality.

- The input to the TM is enclosed by special markers '#'. The TM scans the input at least once, and begins as well as halts with its input head scanning the marker '#' which is to the left of the input.
- The TM has one initial state, one final state, and a set of special states called *transit* states.
- During the computation on any input, whenever the TM moves the input head, it enters one of the transit states. Also, it never enters any of the transit states without moving the input head.

For the purpose of the definitions below, let M be a 2DFA TM with k states, computing a total function. Since M computes a total function, it is clear that on any input x, M can scan $x[i]$, $1 \leq i \leq |x|$, at most k times.

Definition 1 *A* configuration *of M is a 3-tuple $C = \langle st,\ inh,\ outh \rangle$ where st denotes the state of M, inh denotes the input head position and $outh$ denotes the output head position.*

A transit *configuration of M is a configuration whose state is a transit state.*

We shall need to argue about the behavior of a 2DFA TM at the boundary of two adjacent substrings of the input. This behavior is determined by the transit configurations of the TM when it crosses the boundary between the substrings. We identify these configurations below.

Definition 2 *Let u be a substring of x, $u = x[i]x[i+1]\cdots x[j]$. On input x, a* left boundary configuration of M at u *is a transit configuration of M during its computation on x, in which the input head is either (1) at the i^{th} bit and in the previous configuration of M, at the $(i-1)^{th}$ bit, or (2) at the $(i-1)^{th}$ bit and in the previous configuration of M, at the i^{th} bit.*

Similarly, a right boundary configuration of M at u *is a transit configuration of M during its computation on x, in which the input head is either (1) at the j^{th} bit* *and in the previous configuration of M, at the $(j+1)^{th}$ bit, or (2) at the $(j+1)^{th}$ bit and in the previous configuration of M, at the j^{th} bit.*

The following lemma shows that there are two strings such that the behavior of M at their boundaries is identical.

Lemma 1 *Let d_0 be a constant such that $2^{d_0} \geq 2 \cdot (2 \cdot k^2 \cdot d_0)^{2k}$. For the 2DFA TM M, there exist two strings u_0, v_0, $u_0 \neq v_0$, $|u_0| = |v_0| = d_0$, such that for any two strings w_1 and w_2, the sequence of boundary configurations of M at u_0 on the input $w_1 u_0 w_2$ is the same as the sequence of boundary configurations of M at v_0 on the input $w_1 v_0 w_2$.*

Proof. For a string u, $|u| = d_0$, the TM M can 'enter' the substring u of the input $w_1 u w_2$ at most $2k$ times—it can enter u in k different states from the left or from the right. For each of these possibilities, the TM would 'exit' from the substring in one of the k states and either from the left or right. Also, it may output some bits in between. Let S_u be the set of 5-tuples that record the entry-exit behavior of M at u, i.e., $S_u = \{(s_e, p_e, s_x, p_x, o)$, where s_e is one of the k possible entry states, p_e is the entry point (from left or right), and when M enters u in state s_e and from direction p_e, then s_x is the exit state of M, p_x is the exit point of M, and o is the number of bits that the output head advances by during the period M scans u (note that this is independent of the actual position of the output head)$\}$.

Clearly, S_u has exactly $2k$ tuples. Also, the value of o is bounded by $k \cdot d_0$ as the TM can make at most $k \cdot d_0$ steps without exiting from a string of size d_0. So, the number of different S_us is bounded by $(2k^2 \cdot d_0)^{2k}$. Since there are 2^{d_0} strings of size d_0, and $2^{d_0}/(2k^2 \cdot d_0)^{2k} \geq 2$, there must be two strings u_0, v_0 such that $S_{u_0} = S_{v_0}$. Now, for any w_1, w_2, the sequences of boundary configurations of M at u_0 and v_0 will be identical on input $w_1 u_0 w_2$ and $w_1 v_0 w_2$ respectively. □

We now define the notion of a forgetful 2DFA TM.

Definition 3 *A 2DFA TM is* forgetful *if for every n, the sequences of transit configurations of M on inputs of size n are identical. A 2DFA function f is* forgetful *if it can be computed by a forgetful 2DFA TM.*

A forgetful 2DFA TM does not 'remember' the input—except possibly the currently scanned bit—during any stage of the computation. This severely limits the power of the TM. However, we shall show that for any class that is closed under log-lin reduction, any set that is complete for the class under 2DFA reductions is also complete under forgetful 2DFA functions. First, we prove a technical lemma regarding composition of 2DFA functions.

We shall need to compose two 2DFA functions and shall require the composition to remain a 2DFA function. As this may not hold in general, we define the conditions under which the composition is a 2DFA function. Say that a 2DFA function g is a *simple* 2DFA function if there exist four strings b_0, b_1, b_i, and b_f such that for all x, $g(x) = b_i b_{x[1]} b_{x[2]} \cdots b_{x[|x|]} b_f$.

Lemma 2 *Let f be a 2DFA function computed by the TM M, and g be a simple 2DFA function. Then, the function $f \circ g$ is also a 2DFA function. Further, if f is a forgetful 2DFA function, then $f \circ g$ is also forgetful.*

Proof. The following TM M' computes $f \circ g$:

> On input x, start the simulation of M on $g(x)$ by writing the string $\#b_i$ on the work tape. As and when M needs more bits of $g(x)$, look up the corresponding bit of x and write the appropriate string b_0, b_1, $\#b_i$ or $b_f\#$ on the work tape (overwriting the earlier string). (Note that the strings $\#b_i$ and $b_f\#$ are written when the input head of the TM scans the left and right $\#$ respectively.) Output any bit that M outputs.

The TM M' needs only a constant amount of worksapce. A TM that needs a constant amount of workspace can be replaced by a TM that needs no workspace and does all the computation in its states. Thus, the function $f \circ g$ is 2DFA function.

Any time the TM M' moves its input head, its state need only record the current state of M. And so, if M is a forgetful 2DFA TM, then M' is also a forgetful 2DFA TM. □

Theorem 1 *Let $\mathcal{C}$ be any class closed under log-lin reductions. Any set A that is hard for $\mathcal{C}$ under 2DFA reductions is also hard for $\mathcal{C}$ under forgetful 2DFA reductions.*

Proof. Let A be a $\leq_m^{2dfa}$-hard set for $\mathcal{C}$ and B be an arbitrary set in $\mathcal{C}$. We shall exhibit a reduction of B to A computed by a forgetful 2DFA TM. We first define a set D as accepted by the following procedure.

> Input z. Let $z = y10^b$. If b does not divide $|y|$ then reject. Otherwise, let $y = w_0w_1w_2 \ldots w_n$ where $|w_i| = b$ for $0 \leq i \leq n$. Define a new string x, $|x| = n$, with $x[i] = 1$ if $w_i = w_0$, 0 otherwise, for $1 \leq i \leq n$. Accept iff $x \in B$.

It is easy to see that if $B \neq \emptyset, \Sigma^*$, D reduces to B via a log-lin reduction, and if $B = \emptyset$ or Σ^*, D reduces to any set in $\mathcal{C} - \{\emptyset, \Sigma^*\}$ via a log-lin reduction. Therefore, $D \in \mathcal{C}$. Let f be a 2DFA reduction of D to A computed by the TM M with k states. We define a reduction, g, of B to D based on the TM M as given by the following procedure.

> Input x. Let u_0 and v_0 be the two strings of size d_0 that are identified in Lemma 1 for the TM M. Output $v_0w_1w_2 \ldots w_{|x|}10^{d_0}$ where $w_i = v_0$ if $x[i] = 1$, u_0 otherwise, for $1 \leq i \leq |x|$.

It is clear, from the definition of the set D that g is a reduction of B to D. Also, it is easily seen that g is a simple 2DFA reduction. Therefore, $h = f \circ g$ is a 2DFA reduction of B to A by Lemma 2. Further, the TM M' computing h, as defined in the proof of Lemma 2, can be shown to be a forgetful TM as follows. The TM M', on any input of size n, simulates M on the string $v_0w_1 \cdots w_n10^{d_0}$ with w_is as defined above. Using Lemma 1, it can be shown that during this simulation, the state of M at the boundary of any of w_is is independent of its value as $w_i \in \{u_0, v_0\}$. Since M' moves the input head only when M is at the boundary of one of these strings, it follows that M' is forgetful. □

For complete sets under forgetful 2DFA reductions, following ideas from Ref. [6], we can show that they are also complete under size-increasing reductions that are one-one on Σ^n for every $n \geq 1$. However, it does not provide us with the desired result as there can be two strings of *different* lengths, on which the reduction is not one-one. To get around this problem, we need to have an even stronger notion than forgetful TMs.

Definition 4 *A* right scan *of the TM M on some input is the period during which the input head of the TM moves from the marker on the left of the input to the marker on the right. Similarly, a* left scan *is the period during which the input head moves from the marker on the right to the marker on the left. We adopt the convention that a right scan ends the moment input head reaches the marker on the right, and then the next (left) scan begins. Similarly, a left scan ends the moment input head reaches the marker on the left, and then the next (right) scan begins (except when the input head reaches the left marker for the last time).*

The TM M, since it is a 2DFA TM with k states computing a total function, can make at most k left and right scans of the input.

Definition 5 *A 2DFA DTM is* completely forgetful *if for every n, (1) the sequences of transit configurations of M on inputs of size n are identical, and (2) during any scan (left or right) of the input, the states in the transit configurations of M are also identical. A 2DFA function f is* completely forgetful *if it is computed by a completely forgetful 2DFA TM.*

Theorem 2 *Let $\mathcal{C}$ be any class closed under log-lin reductions. Any set A that is hard for $\mathcal{C}$ under 2DFA reductions is also hard for $\mathcal{C}$ under completely forgetful 2DFA reductions.*

Proof. Let A be a $\leq_m^{2dfa}$-hard set for $\mathcal{C}$ and B be an arbitrary set in $\mathcal{C}$. We shall exhibit a reduction of B to A computed by a completely forgetful 2DFA TM. We first define a set E as accepted by the following procedure.

> Input z. Let $z = 0^a1^b0^by0^b1^b0^c$ for some a, b, and c. If b does not divide $|y|$ then reject. Otherwise, let $y = w_1w_2\dots w_n$ where $|w_i| = b$ for $1 \leq i \leq n$. Define a new string x, $|x| = n$, with $x[i] = 0$ if $w_i = 0^b$, 1 otherwise, for $1 \leq i \leq n$. Accept iff $x \in B$.

Set E can clearly be reduced to B via a log-lin reduction (or to some other set in $\mathcal{C}$ if $B = \emptyset$ or Σ^*) and therefore, $E \in \mathcal{C}$. By Theorem 1, we have that E reduces to A via a forgetful 2DFA reduction, say f. Let f be computed by the forgetful 2DFA TM M with k states. We shall define a reduction g of B to E such that $f \circ g$ is a completely forgetful reduction of B to A.

Function g will be a simple 2DFA function with $b_0 = 0^b$, $b_1 = 0^k1^{b-2k}0^k$, $b_i = 0^a1^b0^b$, and $b_f = 0^b1^b0^c$ for some a, $b > 2k$, and c to be specified later. It is easy to see that g is a reduction of B to D. The string $g(x)$ can be divided into $|x|+6$ blocks of consecutive bits—the first block with a bits, the next $|x|+4$ blocks with b bits each, and the last block with c bits. The numbers a, b and c are chosen in such a way that the TM M, on input $g(x)$ and for any scan of the input, is in the *same* state when it crosses the boundary between any two of these blocks for the *last* time during the scan.

Claim *There exist numbers a, $b > 2k$, and c such that for any $n \geq 0$, and for any scan of the input $0^{a+(n+4)\cdot b+c}$, the state of the TM M is the same when it crosses, for the last time during the scan, the boundary between any two blocks of the input, where the blocks are defined as—the first block has a bits, the next $n+4$ blocks have b bits, and the last block has c bits.*

Proof of Claim. The proof of the claim is by induction on the number of the scans of the TM. The induction hypothesis: For any $l \geq 0$, there exist numbers $a_l \geq k$, $b_l > 2k$, and $c_l \geq k$, such that for any $n \geq 0$, and for any of the first l scans of the input $0^{a_l+(n+4)\cdot b_l+c_l}$, the state of the TM M is the same when it crosses, for the last time during the scan, the boundary between any two blocks of the input, where the blocks are defined as—the first block has a_l bits, the next $n+4$ blocks have b_l bits, and the last block has c_l bits.

Base step ($l = 0$): Let $a_0 = k = c_0$, and $b_0 = 2k + 1$. Since the TM has not made any scan of the input yet, the hypothesis trivially holds.

Induction step: Let the hypothesis be true for l. Consider the computation of M on the input $0^{a_l+k\cdot b_l+c_l}$. We know, by the induction hypothesis, that for any of the first l scans of the input, the state of the TM is the same when it crosses, for the last time during the scan, the boundaries between any two of the $k+2$ blocks in the input. Consider the $(l+1)^{th}$ scan. Suppose that it is a right scan. Note that, since the input consists of only zeroes and M has only k states, either the TM never crosses the first k bits of the input, or it reaches the marker on the right. In the first case, since $a_l \geq k$, the TM never reaches even the boundary between the first two blocks and thus the hypothesis trivially holds.

So, let us assume that the TM reaches the marker on the right. Let the states of the TM, when it crosses, for the last time during the $(l+1)^{th}$ scan, the boundaries between the $k+2$ blocks of the input, be $q_1, \ldots, q_{k+1}$. Since the TM has k states, at least two of these states, say q_i and q_j, $i < j$, must be the same. We call the state q_i as the *fixed* state for the $(l+1)^{th}$ scan. Define $a_{l+1} = a_l + (i-1)\cdot b_l$, $b_{l+1} = (j-i)\cdot b_l$, and $c_{l+1} = c_l$. It is easy to verify that for these values, the hypothesis holds for $l+1$.

The proof for the case when the $(l+1)^{th}$ scan is a left scan is very similar (in that, the value of a_{l+1} remains the same while b_{l+1} and c_{l+1} are incremented). □

The above claim will, in fact, hold for any string in $\Sigma^{a+(n+4)\cdot b+c}$ since the TM M is forgetful. Now, the completely forgetful TM M' described below computes the function $f \circ g$. TM M' has, in its finite control, the numbers a, b, c, k and the fixed states of M for every scan (as identified above) written.

> On input x, M' simulates M on $g(x)$. It remembers which scan of M is currently being simulated. At the beginning of the simulation of a right scan of M, the input head of M' would be at the left marker of the input. M' then writes the string $\#0^a1^b0^b$ on the work tape and starts the simulation of M on this string.

During the simulation of this scan, M' moves its input head only when M is in the fixed state for the scan and the input head of M is to the right of the string written on the work tape. This ensures that, during the simulation of M on any string written on the work tape, the input head of M never moves to the left of the string. And when the input head of M moves to the right of the string, say u, M' checks if the state of M is the fixed state for this scan. If it is, M' moves its input head to the right, and depending on the bit written on the new cell, 0 or 1, writes 0^b or $0^k 1^{b-2k} 0^k$ on the work tape (overwriting the previous string) and continues with the simulation. In case M' reads the right end marker, it signals the end of the current scan.

On the other hand, if the state of M is not the fixed state for the current scan, M' writes the string 0^k *after* u and simulates M on this string *without* moving its input head. This simulation is correct since the first k bits of all the three strings 0^b, $0^k 1^{b-2k} 0^k$, and $0^b 1^b 0^c\#$ are zeroes. The crucial point to note here is that the input head of M cannot go beyond $u0^k$ without returning to the right boundary of u. This is because M must cross the right boundary of the string u at least once more during the scan (since M is not yet in the fixed state for the scan), and this it cannot do if its input head goes k bits away from u (there being only k states of M). Thus, M would eventually cross the right boundary of u in the fixed state for the scan. Now, M' works as above.

A left scan of M can be similarly simulated. During the entire simulation, M' outputs any bit that M does.

M' uses only a constant amount of workspace, and thus can be converted to a 2DFA TM. It is also a completely forgetful 2DFA TM since during any scan of the input, it always moves the input head in one direction only, and whenever it moves the input head, it only needs to records the current scan number in its finite control apart from the information that is scan-independent. □

4. The Structure of $\leq_m^{2dfa}$-Complete Degrees

In this section we prove our main results, viz., for every class $\mathcal{C}$ closed under log-lin reductions, the $\leq_m^{2dfa}$-complete sets for $\mathcal{C}$ are also $\leq_{1,li}^{2dfa}$-complete but not 2DFA-isomorphic.

We first give a useful property of completely forgetful reductions.

Lemma 3 *Let f be a completely forgetful reduction. Then there exists an even number $l \geq 0$, and strings O_0^i, O_1^i, E^i, E^{l+1} with $|O_0^i| = |O_1^i|$ for $1 \leq i \leq l$, such that for every x,*

$$\begin{aligned} f(x) &= E^1 O^1_{x[1]} O^1_{x[2]} \cdots O^1_{x[|x|]} E^2 O^2_{x[|x|]} O^2_{x[|x|-1]} \cdots O^2_{x[1]} E^3 \cdots\cdots \\ &\qquad \cdots\cdots E^{l-1} O^{l-1}_{x[1]} O^{l-1}_{x[2]} \cdots O^{l-1}_{x[|x|]} E^l O^l_{x[|x|]} O^l_{x[|x|-1]} \cdots O^l_{x[1]} E^{l+1}. \end{aligned}$$

Proof. Let M be a completely forgetful 2DFA TM computing f. By definition, we have that during any scan of the input tape, the TM M is in the same state in any transit configuration, and the transit configuration is independent of the value of the bit just scanned. This implies that (1) the TM makes the same number of scans on any input, (2) the output of the TM while it scans a bit during any scan is independent of the position of the bit, and (3) the number of bits output on reading a bit is independent of the value of the bit. Let l be the number of scans the TM makes on any input, O_0^i and O_1^i be the output of M on reading 0 and 1 respectively during the i^{th} scan. The TM may also output some bits while scanning the markers at the two ends of the input. Let E^i denote the output of M while scanning the marker # at the beginning of the i^{th} scan for $1 \leq i \leq l$ (if i is odd, the TM scans the marker on the left, otherwise the marker on the right), and E^{l+1} denote the output of M while scanning the left marker at the end of the computation. It is straightforward now to see that for any string x, $f(x)$ is as described above. □

Theorem 3 *For any class $\mathcal{C}$ closed under log-lin reductions, $\leq_m^{2dfa}$-hard sets for $\mathcal{C}$ are also $\leq_{1,li}^{2dfa}$-hard.*

Proof. Let A be a $\leq_m^{2dfa}$-hard set for $\mathcal{C}$. Let $B \in \mathcal{C}$. We show that B is reducible to A via a one-one, size-increasing, 2DFA reduction. Define a set F as:

$$F = \{1x \mid x \in B\} \cup \{1\}.$$

The set F is clearly in $\mathcal{C}$. Let f be a completely forgetful reduction of F to A computed by the TM M (the existence of such a function is guaranteed by Theorem 2). By Lemma 3, we have that for any y, $f(y)$ can be written as:

$$\begin{aligned} f(y) \quad = \quad & E^1 O^1_{y[1]} O^1_{y[2]} \cdots O^1_{y[|y|]} E^2 O^2_{y[|y|]} O^2_{y[|y|-1]} \cdots O^2_{y[1]} E^3 \cdots\cdots \\ & \cdots\cdots E^{l-1} O^{l-1}_{y[1]} O^{l-1}_{y[2]} \cdots O^{l-1}_{y[|y|]} E^l O^l_{y[|y|]} O^l_{y[|y|-1]} \cdots O^l_{y[1]} E^{l+1}, \end{aligned} \quad (1)$$

for some l, O_0^i, O_1^i, E^i, E^{l+1}, $1 \leq i \leq l$. Also, we have that $|O_0^i| = |O_1^i|$. So,

$$|f(y)| \quad = \quad |y| \cdot (\sum_{i=1,l} |O_1^i|) + \sum_{i=1,l+1} |E^i|. \quad (2)$$

Suppose that for every i, $1 \leq i \leq l$, $O_0^i = O_1^i$. Then, by (1), for every n, and for every pair of strings y_1 and y_2 with $|y_1| = |y_2| = n$, $f(y_1) = f(y_2)$. In particular, $f(1) = f(0)$. However, this is not possible since $1 \in F$, $0 \notin F$ and f is a reduction of F to A. Therefore, there exists an i, $1 \leq i \leq l$, such that $O_0^i \neq O_1^i$. It follows that $\sum_{i=1,l} |O_1^i| \geq 1$. Therefore, by (2), $|f(y)| \geq |y|$ for every y.

For any two strings y_1 and y_2, $y_1 \neq y_2$, if $|y_1| = |y_2|$, then by (1), $f(y_1) \neq f(y_2)$. And if $|y_1| > |y_2|$, then by (2), $|f(y_1)| - |f(y_2)| = (|y_1| - |y_2|) \cdot (\sum_{i=1,l} |O_1^i|) \geq 1$ since $\sum_{i=1,l} |O_1^i| \geq 1$. Therefore, f is one-one.

Let $g(x) = 1x$. It is obvious that g is a one-one, size-increasing, simple 2DFA reduction of B to F, and the function $h \circ g$ is a completely forgetful 2DFA reduction of B to A. Since both f and g are one-one and size-increasing, h is also one-one and size-increasing. This complete the proof of the theorem. □

Corollary 1 *For any class $\mathcal{C}$ closed under log-lin reductions, $\leq_m^{2dfa}$-complete sets for $\mathcal{C}$ are also $\leq_{1,li}^{2dfa}$-complete.*

Recall that *first-order functions* are functions computed by DLOGTIME-uniform AC^0 circuits.[11]

Corollary 2 *For any class $\mathcal{C}$ closed under log-lin reductions, $\leq_m^{2dfa}$-complete sets for $\mathcal{C}$ are first-order isomorphic.*

Proof. By Theorem 2, any $\leq_m^{2dfa}$-complete set for $\mathcal{C}$ is also complete under completely forgetful 2DFA reductions. Any completely forgetful 2DFA function f has the structure described in Lemma 3. Any such function is actually a *projection*[13] since every bit of the output depends on at most one bit of the input. In Ref. [14], it is shown that any two sets complete for a 'nice' complexity class under *first-order projections*[15] (these are a uniform version of projections) are first-order isomorphic, and their proof also works for any class closed under log-lin reductions. So, all we need to show is that a completely forgetful 2DFA function is actually a first-order projection.

Let f be a completely forgetful 2DFA function computed by the TM M, and $\mathrm{BIT}(i)$ be a unary predicate that is 1 iff the i^{th} bit of the input to f is 1. For the output of f, we define a predicate $\mathrm{OUT}(i,r,a)$ which denotes the a^{th} bit of the output of M while scanning the i^{th} bit of the input during the r^{th} scan of the input if $i > 0$; and the a^{th} bit of the output of M while scanning the corresponding # marker during the r^{th} scan if $i = 0$. In other words, $\mathrm{OUT}(i,r,a)$, for an input of size n, is the a^{th} bit of the string $O^r_{\mathrm{BIT}(i)}$ if $1 \leq i \leq n$, and the a^{th} bit of the string E^r if $i = 0$. Note that we must allow r to go up to $l+1$ when $i = 0$ to take care of the string E^{l+1}. It is straightforward to see that, depending on the values of a and r (which are only a constantly many), the value of $\mathrm{OUT}(i,r,a)$ is either fixed, or determined in a fixed way by $\mathrm{BIT}(i)$. Thus, $\mathrm{OUT}(i,r,a)$ can be written as a finite disjunction of conjuncts with each conjunct having at most one occurrence of the BIT predicate. This shows that f is a first-order projection.[14] $\square$

Though there are no $\leq_m^{2dfa}$-complete sets for the classes P, NP, PSPACE etc., the classes $\mathrm{DSPACE}(n^k)$, $\mathrm{NSPACE}(n^k)$, E, NE etc. are closed under log-lin reductions and have $\leq_m^{2dfa}$-complete sets. For example, the following set is $\leq_m^{2dfa}$-complete for $\mathrm{DSPACE}(n^k)$.

$$K = \{i \mid \text{ DTM } M_i \text{ accepts } i \text{ within space } k \cdot |i|^k + k\}.$$

So, we have,

Corollary 3 *For any class $\mathcal{C}$, $\mathcal{C} \in \{\mathrm{DSPACE}(n^k), \mathrm{NSPACE}(n^k), \mathrm{E}, \mathrm{NE}\}$, $\leq_m^{2dfa}$-complete sets for $\mathcal{C}$ are also $\leq_{1,li}^{2dfa}$-complete, and are first-order isomorphic to each other.*

Corollaries 1 and 2, while proving a strong collapse of $\leq_m^{2dfa}$-complete degrees, do not show that these degrees collapse to 2DFA-isomorphic degrees. We now show that such a collapse is not possible.

We first translate the notion of annihilating functions defined in Ref. [5] to our settings. A function f is a *2DFA-annihilating* function if f is one-one, length-increasing and all 2DFA-computable subsets of the range of f are sparse.

Lemma 4 *Function $t(x) = xx$ is a 2DFA-annihilating function.*

Proof. Function t is clearly a one-one, length-increasing 2DFA function. Let a 2DFA TM M^* recognize a subset R of its range. We show that R has only a constantly many strings of every length. For any n, and for any x of length n, consider the sequence of right boundary configurations of M^* at the first x of the input $f(x) = xx$. There are at most k such configurations where k is the number of states of M^*. Therefore, the number of different such sequences of configurations is bounded by k^k. If two strings xx and yy in R have the same sequence of configurations as above, then xy and yx will also have the same sequence. This implies that strings xy and yx also belong to R which is not possible if $x \neq y$ as R is a subset of the range of t. Hence R has at most k^k strings of length n, for any n. □

Theorem 4 *For every class $\mathcal{C}$ closed under log-lin reductions, the 2DFA-isomorphism conjecture is false.*

Proof. Consider a $\leq_m^{2dfa}$-complete set A for $\mathcal{C}$. Define,

$$B = \{x \mid (x = 1y \wedge y \in A) \vee x = 0y\}.$$

The set B reduces to A via a log-lin reduction and is therefore in $\mathcal{C}$. It is also $\leq_m^{2dfa}$-complete for $\mathcal{C}$ since any 2DFA reduction to A can be transformed to a 2DFA reduction to B by just appending a '1' at the beginning of its output.

Let $t(x) = xx$, and consider the set $t(B)$. This set also belong to $\mathcal{C}$ as it can be reduced to B via a log-lin reduction. It is also $\leq_m^{2dfa}$-complete for $\mathcal{C}$ since any 2DFA reduction to B can be transformed to a 2DFA reduction to $t(B)$ by making it output the same string twice. Suppose that B is 2DFA-isomorphic to $t(B)$ via h. So, $B \leq_{1,i}^{2dfa} t(B)$ via h. A 2DFA TM that accepts any input whose image under h^{-1} begins with a zero recognizes a dense subset of the set $t(B)$. This contradicts Lemma 4 since $t(B)$ is contained in the range of t. □

5. Concluding Remarks

Theorem 4 shows that the *2DFA-encrypted complete set conjecture* is true for any class $\mathcal{C}$ closed under log-lin reductions. This conjecture was proposed by Joseph and Young[4] contradicting the isomorphism conjecture. In Ref. [6], the *r-encrypted complete set conjecture for class $\mathcal{C}$* was defined as: there is a $\leq_m^r$-complete set A and an r-computable one-one, length-increasing function f such that $A \not\leq_{1,li,i}^r f(A)$. As shown in Theorem 4, $B \not\leq_{1,li,i}^{2dfa} t(B)$ for the function $t(x) = xx$.

Say that (1) reducibility r is *simple* for class $\mathcal{C}$ if $\leq_m^r$-complete sets for $\mathcal{C}$ are also $\leq_{1,li}^r$-complete; and (2) reducibility r is *deterministically invertible* for class $\mathcal{C}$ if—for *every* resource bound s, $s \geq r$, such that the inverses of one-one, length-increasing r-computable functions can be computed by non-deterministic TMs within a resource bound of s—the $\leq_{1,li}^r$-complete sets for $\mathcal{C}$ are also $\leq_{1,li}^r$-complete via reductions whose inverses are computable by deterministic TMs within a resource bound of s.

In Ref. [6], the *r-complete degree conjecture for class $\mathcal{C}$* was proposed stating that reducibility r is both simple and deterministically invertible for the class $\mathcal{C}$. It was shown there that if the p-complete degree conjecture is true for a class $\mathcal{C}$ then the p-isomorphism conjecture is also true for the class $\mathcal{C}$. Further, it was shown that for several weak reducibilities, including 1-L and 1-NL, the complete degree conjecture holds for all classes closed under log-lin reductions.

Does the 2DFA-complete degree conjecture also hold for classes closed under log-lin reductions? It appears so. Firstly, 2DFA reducibility is simple for any class closed under log-lin reductions (follows from Corollary 1). Moreover, the inverses of one-one, length-increasing 2DFA reductions can be computed by *mu-NFA* TMs where a mu-NFA TM is a multihead, nondeterministic TM with a read-only input tape, a write-only output tape such that each of its input heads is unidirectional, i.e., it either moves left or right. The input heads are also assumed to be 'sensitive', i.e., a head can detect whether any other head is scanning the same input cell as itself.

Proposition Any one-one, length-increasing 2DFA function is mu-NFA-invertible.

Proof Sketch. Let M be a 2DFA TM computing a function g. For an input string y, we now describe a multihead, unidirectional NFA TM $\widehat{M}$ computing $x = g^{-1}(y)$. Let the number of states of M be k. Then, M can visit any input cell at most k times. Each input cell can be visited from left or right corresponding to whether the previously visited cell was to the right or left of the current cell. $\widehat{M}$ has $2k$ heads—k left moving and k right moving. The left moving heads are for the left visits to cells and the right moving ones are for the right visits. The output of M on some cell is defined to be the output produced while M is scanning that cell.

$\widehat{M}$ initially guesses the positions in the input y of the output of M as well as the states of M during each scan of $x[1]$. It places a right moving head each at the positions corresponding to the output of M on $x[1]$ during a right visit, and a left moving head each at the positions corresponding to the output during a left visit. Now it guesses the bit $x[1]$ and verifies if the output of M agrees with the bits written at the guessed positions. For this, $\widehat{M}$ has to guess the previous states for the left moving heads and verify that these previous states lead M to the current states on $x[1]$. For right moving heads, $\widehat{M}$ just needs to compute the next state of M. If the output of M agrees then $\widehat{M}$ outputs the guessed bit $x[1]$ and proceeds with the simulation otherwise aborts.

Now, $\widehat{M}$ guesses the bit $x[2]$ and does the same verification as above. Note that it is possible that M may have made a 'turn' at $x[2]$ without touching $x[1]$ at some time. So, $\widehat{M}$ also guesses these 'turning' points and corresponding states for the verification for $x[2]$. In this way the simulation is continued. Whenever a right moving head meets with a left moving head (this corresponds to the situation when M makes a 'turn'), $\widehat{M}$ checks if their corresponding states are the same. If yes, then it continues with the simulation freezing the two heads (no more simulation needs to be done for these two heads), otherwise aborts.

It is easy to verify that the above simulation computes $g^{-1}(y)$ correctly. □

In fact, there appears to be no better way of computing the inverses of arbitrary one-one, length-increasing 2DFA functions—Lemma 4 already provides some evidence for this by showing that $t(x) = xx$ cannot be inverted by a 2DFA TM. Moreover, it is easy to see that the function $f(x) = xx^R$ where x^R is the reverse of string x, cannot be computed by one-way, multihead DTMs without workspace.

By modifying the construction of the set F and the function g in the proof of Theorem 3 a little bit, one can show that the $\leq_m^{2dfa}$-complete sets for any class closed under log-lin reductions are also complete under one-one, size-increasing reductions whose inverses can be computed by *mu-DFA* TMs which are deterministic versions of mu-NFA TMs. Thus, the 2DFA reducibility appears to be deterministically invertible for any class closed under log-lin reductions implying that the 2DFA-complete degree conjecture for these classes is likely to be true.

Acknowledgements

The authors wish to thank the two anonymous referees whose thoughtful comments helped in improving the paper.

References

1. L. Berman and J. Hartmanis, "On isomorphism and density of NP and other complete sets," *SIAM J. Comput.* **1** (1977) 305–322.
2. J. Myhill, "Creative sets," *zeitschrift für Mathematische Logic und Grundlagen der Mathematik* **1** (1955) 97–108.
3. S. Kurtz, S. Mahaney and J. Royer, "The structure of complete degrees," in *Complexity Theory Retrospective*, ed. A. Selman (Springer-Verlag, 1988) pp. 108–146.
4. D. Joseph and P. Young, "Some remarks on witness functions for nonpolynomial and noncomplete sets in NP," *Theoretical Computer Science* **39** (1985) 225–237.
5. S. Kurtz, S. Mahaney and J. Royer, "The isomorphism conjecture fails relative to a random oracle," in *Proc. Symposium on Theory of Computation*, 1989, pp. 157–166.
6. M. Agrawal, "On the isomorphism problem for weak reducibilities," to appear in *J. Comput. System Sci.* A preliminary version appeared in *Proc. 9th Structure in Complexity Theory Conference*, Amsterdam, Jun. 1994, pp. 338–355.
7. L. J. Stockmeyer, "The Complexity of decision problems in automata theory and logic," Ph. D. Thesis, Massachusetts Institute of Technology, 1974.
8. E. W. Allender, "Isomorphisms and 1-L reductions," *J. Comput. System Sci.* **36** (1988) 336–350.
9. K. Ganesan and S. Homer, "Complete problems and strong polynomial reducibilities," *SIAM J. Comput.* **21** (1992) 733–742.
10. M. Agrawal and S. Biswas, "Polynomial isomorphism of 1-L-complete sets," to appear in *J. Comput. System Sci.* A preliminary version appeared in *Proc. 8th Structure in Complexity Theory Conference*, San Diego, May 1993, pp. 75–80.
11. D. Barrington, N. Immerman and H. Straubing, "On uniformity within NC^1," *J. Comput. System Sci.* **74** (1990) 274–306.
12. J. Hopcroft and J. D. Ullman, *Introduction to Automata, Languages and Computation* (Addison-Wesley, 1979).
13. L. G. Valiant, "Reducibility by algebraic projections," *L'Enseignement mathématique* **28** (1982) 253–268.
14. E. Allender, J. Balcázar and N. Immerman, "A first-order isomorphism theorem," to appear in *SIAM J. Comput.* A preliminary version appeared in *Proc. 10th Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science 665, 1993, pp. 163–174.

15. N. Immerman and S. Landau, "The complexity of iterated multiplication," *Information and Computation* **116** (1995) 103–116.