CS294-152: Lower Bounds August 27, 2018

Time-Space Lower Bounds and the Relativization Barrier

Instructor: Ryan Williams *Scribe: Siqi Liu*

1 Introduction

In this lecture, we talk about time-space tradeoffs for the Satisfiability problem (SAT). Though it remains unclear whether SAT has a logarithmic-space algorithm, it is known that for sufficiently small constant c, SAT does not have $O(n^c)$-time $O(\log n)$-space algorithms. We will see a proof of this result using the quantifier-trading technique. We will also introduce the Relativization Barrier, and give an oracle relative to which our SAT lower bound is false.

2 Preliminaries

In this section we define a few complexity classes that will be used in the theorem statements and proofs in Section 3 and 4.

Recall a decision problem is a function $f : \{0,1\}^\star \to \{0,1\}$.

Definition 1 *LOGSPACE is the class of decision problems that can be decided by some Turing machine $\mathcal{M}$ that uses at most $O(\log n)$ extra worktape (beyond its input, which is read-only).*

Definition 2 *coNTIME$[t(n)]$ is the class of decision problems for which there exists a $O(n)$-time TM $\mathcal{M}$ such that for every $x \in \{0,1\}^*$,*

$$x \in L \equiv \forall z \in \{0,1\}^{O(t(|x|))}, \mathcal{M}(x,z) = 1.$$

We define NTIME$[t(n)]$ analogously, but with an $\exists$ instead of a $\forall$.

Definition 3 $NP = \cup_{k\geq 0} NTIME[n^k]$

Definition 4 *TS$[t(n), s(n)]$ is the class of decision problems that can be decided by some $O(t(n))$-time Turing machine $\mathcal{M}$ that uses $O(s(n))$ extra work space (beyond the read-only input).*

Definition 5 *$\sum_2$ TIME$[t(n)]$ denotes the class of decision problems for which there exists a $O(n)$-time Turing machine $\mathcal{M}$ such that for every $x \in \{0,1\}^*$,*

$$x \in L \iff \exists y \in \{0,1\}^{O(t(|x|))} \forall z \in \{0,1\}^{O(t(|x|))}, \mathcal{M}(x,y,z) \text{ accepts.}$$

3 Time-space Lower Bounds for SAT

It is not hard to see that LOGSPACE $\subseteq P$: For any $L \in$ LOGSPACE, there exists a TM $\mathcal{M}$ that uses $O(\log n)$ locations on the work tape on inputs of length n. Thus there are at most $2^{O(\log n)}$

possible "configurations" that $\mathcal{M}$ can be in, on inputs of length n. Therefore, on an input of length n, $\mathcal{M}$ either terminates within $poly(n)$ steps on any input, or is in an infinite loop (in which case it will not accept).

However, a couple of related questions are wide open: LOGSPACE $= P$? Or even LOGSPACE $=$ NP? Answers for these questions are not known yet, but it is widely believed that LOGSPACE $\neq P \neq$ NP. Towards figuring out whether LOGSPACE $\neq$ NP, we start by trying to answer the following time-space tradeoff question for SAT: *Given $O(\log n)$ space to solve SAT, does it require more than $O(n^k)$ time?*

If the answer is affirmative for all $k \geq 0$, we can infer that LOGSPACE $\neq$ NP. On the other hand if LOGSPACE $=$ NP, we would get the following 3 statements:

(1) NP $=$ LOGSPACE

(2) NTIME$[n] \subseteq$ LOGSPACE

(3) SAT $\in$ LOGSPACE

Claim 6 *(1) $\equiv$ (2) $\equiv$ (3)*

Proof: Here we sketch a few interesting parts of the proof.

(1) $\equiv$ (2): $\Rightarrow$, NP $= \cup_{k\geq 0}$NTIME$[n^k]$; $\Leftarrow$, by a padding argument if NTIME$[n] \subseteq$ TS$[t(n), \log n]$, then NTIME$[n^k] \subseteq$ TS$[t(n^k), \log n^k] =$ TS$[t(n^k), \log n] \subseteq$ LOGSPACE.

(1) $\equiv$ (3): $\Rightarrow$, SAT $\in$ NP; $\Leftarrow$ SAT is NP-complete under logspace reductions. $\Box$

3.1 Quantifier-Trading Proofs

Though SAT $\notin$ TS$[n^k, \log n]$ is not known to be true for all $k \geq 0$, we know that it is true for $k = 2\cos\frac{\pi}{7} - o(1)$. This result is an immediate corollary of the following theorem.

Theorem 7 (W'07) *For all $\epsilon > 0$, NTIME$[n] \not\subseteq$ TS$[n^{2\cos\frac{\pi}{7}-\epsilon}, \log n]$.*

We will not present the proof of this theorem, but we will prove a weaker result using the quantifier-trading technique. We will prove that:

Theorem 8 (Fortnow'97, FLvMV'00) *For all $c < \sqrt{2}$, NTIME$[n] \not\subseteq$ TS$[n^c, \log n]$.*

Proof: The proof proceeds by contradiction and consists of the following four steps:

(1) Assume that
$$\text{NTIME}[n] \subseteq \text{TS}[n^c, \log n], \text{ for some } c > 1 \quad (1)$$

(2) From the (unconditional) theorem 9, we get
$$\text{TS}[n^c, \log n] \subseteq \sum_2 \text{TIME}[n^{c/2} \log n].$$

(3) From 1, we deduce theorem 10 and infer that

$$\sum_2 \mathrm{TIME}[n^{c/2}\log n] \subseteq \mathrm{NTIME}[n^{c^2/2}\mathrm{poly}\log n].$$

1. (1), (2), and (3) together implies that

$$\mathrm{NTIME}[n] \subseteq \mathrm{NTIME}[n^{c^2/2}\mathrm{poly}\log n].$$

For all $c < \sqrt{2}$, the conclusion contradicts the nondeterministic time hierarchy theorem. So we get a contradiction for $c < \sqrt{2}$.

$\square$

Theorem 9 *Speed Up Theorem: For $t(n) \geq n$ and $s(n) \geq \log n$,*

$$TS[t(n), s(n)] \subseteq \sum_2 TIME[t(n)^{1/2}s(n)].$$

Proof: Think of a computation for some $L \in \mathrm{TS}[t(n), s(n)]$ as a table of width $O(s(n))$ and length $t(n)$, where the string in row i represents the configuration of the TM $\mathcal{M}$ at time step i. Now we define a more general problem: given configurations C and D of a TM $\mathcal{M}$, does $\mathcal{M}$ reach D from C in $t(n)$ steps?

If the answer is yes, then there **exist** $\ell = \lceil t(n)^{1/2} \rceil$ configurations $C_0 = C, C_1, \ldots, C_{\ell-1} = D$ such that **for all** $i = 1, \ldots, \ell$, $\mathcal{M}$ reaches C_{i+1} from C_i within $t(n)^{1/2}$ steps. Then we construct a TM $\mathcal{M}'$ that takes input $C, D, C_0, \ldots, C_{\ell-1}$ and some $z \in \{0, \ldots, \ell-1\}$, which simulates $\mathcal{M}$ to check if $\mathcal{M}$ reaches C_{z+1} from C_z within $t(n)^{1/2}$ steps. If $z = 0$, $\mathcal{M}'$ also checks if $C_0 = C$, and if $z = \ell - 1$, $\mathcal{M}'$ checks if $C_{\ell-1} = D$. It is clear that $\mathcal{M}'$ runs in $O(s(n))t(n)^{1/2}$ time on input of length $O(s(n))t(n)^{1/2}$.

By definition of the class $\sum_2 \mathrm{TIME}[.]$, the existence of $\mathcal{M}'$ implies that $L \in \sum_2 \mathrm{TIME}[t(n)^{1/2}s(n)]$. The theorem follows. $\square$

Theorem 10 *Slow Down Theorem: Assumption* (1) *implies that* $\sum_2 TIME[n^k] \subseteq NTIME[n^{kc}]$.

Proof: Let $f \in \sum_2 \mathrm{TIME}[n^k]$. We are going to show that under assumption (1), $f \in \mathrm{NTIME}[n^{kc}]$. Since $f \in \sum_2 \mathrm{TIME}[n^k]$, if $f(x) = 1$, then there exists $y \in \{0,1\}^{O(n^k)}$ and $O(n)$-time TM $\mathcal{M}$ such that $\mathcal{M}(x, y, z) = 1$, $\forall z \in \{0,1\}^{O(n^k)}$.

Define a new decisional problem g, such that for all $x \in \{0,1\}^n$ and $y \in \{0,1\}^{O(n^k)}$ $g(x, y) = 1$ iff $\mathcal{M}(x, y, z) = 1$, $\forall z \in \{0,1\}^{O(n^k)}$. By definition g is in coNTIME$[n]$. Assuming (1), we deduce that $g \in \mathrm{TS}[n^c, \log n]$.

So there exists a TM $\mathcal{M}'$ that decides g in $O(n^c)$ time. Using $\mathcal{M}'$ we can construct a non-deterministic TM $\mathcal{M}''$ that computes f in $O(n^{kc})$ time in the following way: $\mathcal{M}''$ guesses a y on input x, then runs $\mathcal{M}'$ on (x, y) and outputs the outcome of $\mathcal{M}'(x, y)$. Therefore $f \in \mathrm{NTIME}[n^{kc}]$. The theorem follows. $\square$

Better bounds for $c \geq \sqrt{2}$ can be obtained using the quantifier-trading technique illustrated in the proof above. All known lower bounds proofs for time-space tradeoffs of NTIME$[n]$ can be unified under some common proof system $\mathcal{P}$. However, there is a limit of the proof system as captured in the following theorem.

Theorem 11 (BW'13) *For all $\epsilon > 0$, there is no proof in the proof system $\mathcal{P}$ of the statement "NTIME$[n]$ is not in TS$[n^{2\cos\frac{\pi}{7}+\epsilon}, \log n]$".*

4 Oracles and Relativization: a Blessing and a Barrier

Often a theorem in complexity theory still holds when all the Turing machines or algorithms involved in the theorem have access to a common oracle A. An example is the universal simulation theorem (UST) and the "oracle" UST.

Theorem 12 *Universal simulation theorem: There exists an algorithm U such that for all algorithms $\mathcal{M}$, and inputs x and t, $U(\mathcal{M}, x, t)$ accepts iff $\mathcal{M}(x)$ accepts within t steps.*

Theorem 13 *"Oracle" UST: For all oracles A, there exists an algorithm U^A such that for all algorithms $\mathcal{M}^A$ with oracle A, and inputs x and t, $U^A(\mathcal{M}, x, t)$ accepts iff $\mathcal{M}^A(x)$ accepts within t steps.*

When this generalization holds, we say that the theorem "relativizes" and it holds "relative to every oracle". Here we list a couple more examples.

1. "Relativized" time hierarchy theorem: For all oracles A, there is $f \in \text{TIME}^A[t^c(n)] \backslash \text{TIME}^A[t^n]$ for $c = 1 + o(1)$.
2. "Relativized" non-deterministic time hierarchy theorem: For all oracles A, there is $f \in \text{NTIME}^A[t^c(n)] \setminus \text{NTIME}^A[t^n]$ for $c = 1 + o(1)$.
3. For all oracles A, $P^A \subseteq \text{NP}^A$.

However there is a barrier to this generalization. The barrier comes about when you find two complexity classes C and D such that there exists oracles $A \neq B$, $C^A = D^A$ and $C^B \neq D^B$. This barrier implies that to conclude how C compares to D, we require proof techniques that do not yield relativized theorems. Such a barrier occurs in comparing P and NP.

Theorem 14 *There exists an oracle A such that $P^A = NP^A$.*

This theorem shows that if $P \neq \text{NP}$, we need non-relativizing techniques to prove it.

A similar barrier exists for the problem we studied in the previous section. One can show that the statement $\text{NTIME}[n] \not\subseteq \text{TS}[n^c, \log n]$ for all $c > 0$ is **false** relative to some oracle. We conclude that the techniques we used to prove the lower bound in the previous section "do not relativize" in some sense. (Note: this conclusion is evidently not widely accepted, after conversations with Russell Impagliazzo!)

Theorem 15 *There exists an oracle A, such that $NTIME^A[n] \subseteq TS^A[n^{1.1}, \log n]$.*

Here, we use the (standard) oracle access mechanism for space-bounded computation: the oracle tape is write-only, append-only (its read/write head only moves to the right), and it does not count towards the space bound.

Proof: Let $\mathcal{M}_1^?, \mathcal{M}_2^?, \mathcal{M}_3^?, \ldots$ be an enumeration of all non-deterministic $O(n)$-time machines with an oracle, where each $\mathcal{M}_i^?$ runs in at most $i \cdot n$ steps on inputs of length n. We construct an oracle A in the following way:

For $k = 1, 2 \ldots,$ perform the following procedure, which we call "stage k". For all $i = 1, \ldots, \log(k)$ and all binary strings x of length k, run $\mathcal{M}_i^A$ on x over all computation paths, where we set A to answer "no" on all queries by the $\mathcal{M}_i^A(x)$ that were not asked in stages $1, \ldots, k-1$.

Let $S_k = \{(i, x) \,|\, \mathcal{M}_i^A \text{ accepts } x\}$. Then for all $(i, x) \in S_k$, add the string $0^{k^{1.1} + k\log(k)} 1(i, x)$ to the oracle A. For sufficiently large k, none of these strings have been queried by any $\mathcal{M}_i$ on any x: each $\mathcal{M}_i(x)$ runs in at most $ik \leq k\log(k)$ steps (so it cannot ask queries longer than that), but each $0^{k^{1.1}+k\log(k)}1(i, x)$ has length greater than $k^{1.1} + k\log(k)$. Thus our oracle A is well-defined on each string.

Given the oracle A above, it is easy to simulate any $\mathcal{M}_i^A$ with an oracle for A in time $n^{1.1}$ and space $O(\log n)$: On input x, use an $O(\log n)$-bit counter to write the string $0^{|x|^{1.1}+|x|\log(|x|)}1(i, x)$ on the write-only oracle tape. Then call A on the string and output the answer. The theorem follows. $\square$

References

[1] Sanjeev Arora, and Boaz Barak. *Computational Complexity: A Modern Approach* . Cambridge University Press, New York, 2009.

[2] Samuel R. Buss, and Ryan Williams. *Limits on alternation-trading proofs for time-space lower bounds.* Conference on Computational Complexity, 2013.

[3] Lance Fortnow. *Nondeterministic polynomial time versus nondeterministic logarithmic space.* In proceeding of IEEE Conference on Computational Complexity, 1997.

[4] Lance Fortnow, Richard Lipton, Dieter Van Melkebeek, and Anastasios Viglas. *Time-space lower bounds for satisfiability.* J. ACM, 52:835-865, 2005. Prelim version CCC 2000.

[5] Ryan Williams. *Algorithms and resource requirements for fundamental problems.* 2007.