Computational Complexity Theory
Fall 2025

Time-space lower bounds for SAT
September 9, 2025

Lijie Chen
University of California, Berkeley
lijiechen@berkeley.edu

Motivation: Hardness for NP

- SAT: the canonical NP-complete problem.
- The one million dollar question: Is there a polynomial time algorithm for SAT?
- This is way too hard apparently (e.g., the relativization barrier), can we prove something weaker first?
- This lecture: one of the strongest hardness results for SAT we know.
- **Bonus:** the proof crucially uses Time hierarchy theorem!

Recap: Multi-tape Turing Machine Model

Definition (Multi-tape Turing Machine)

- A multi-tape Turing machine has k tapes for some constant k.

Recap: Multi-tape Turing Machine Model

Definition (Multi-tape Turing Machine)

- A multi-tape Turing machine has k tapes for some constant k.
- Each tape has its own read/write head.

Recap: Multi-tape Turing Machine Model

Definition (Multi-tape Turing Machine)

- A multi-tape Turing machine has k tapes for some constant k.
- Each tape has its own read/write head.
- The input is initially written on the first tape (input tape).

Recap: Multi-tape Turing Machine Model

Definition (Multi-tape Turing Machine)

- A multi-tape Turing machine has k tapes for some constant k.
- Each tape has its own read/write head.
- The input is initially written on the first tape (input tape).
- The machine can read/write symbols and move heads independently on each tape.

Recap: Multi-tape Turing Machine Model

Definition (Multi-tape Turing Machine)

- A multi-tape Turing machine has k tapes for some constant k.
- Each tape has its own read/write head.
- The input is initially written on the first tape (input tape).
- The machine can read/write symbols and move heads independently on each tape.
- One step of computation allows the machine to:
  - Read the current symbols under all heads
  - Write new symbols on all tapes
  - Move each head left, right, or keep it stationary
  - Change the internal state

Space-Bounded Computation

Definition (Space complexity)

- For a multi-tape Turing machine M and input x, the **space complexity** of M on x is the maximum number of tape cells visited by any head during the computation.

Space-Bounded Computation

Definition (Space complexity)

- For a multi-tape Turing machine M and input x, the **space complexity** of M on x is the maximum number of tape cells visited by any head during the computation.
- We say M uses space $S(n)$ if for every input x of length n, M uses at most $S(n)$ space.

Space-Bounded Computation

Definition (Space complexity)

- For a multi-tape Turing machine M and input x, the **space complexity** of M on x is the maximum number of tape cells visited by any head during the computation.
- We say M uses space $S(n)$ if for every input x of length n, M uses at most $S(n)$ space.
- The input tape is **read-only** and does not count towards space usage.

Space-Bounded Computation

Definition (Space complexity)

- For a multi-tape Turing machine M and input x, the **space complexity** of M on x is the maximum number of tape cells visited by any head during the computation.
- We say M uses space $S(n)$ if for every input x of length n, M uses at most $S(n)$ space.
- The input tape is **read-only** and does not count towards space usage.
- There is a designated **output tape** which is **write-only** and does not count towards space usage.

Space-Bounded Computation

Definition (Space complexity)

- For a multi-tape Turing machine M and input x, the **space complexity** of M on x is the maximum number of tape cells visited by any head during the computation.
- We say M uses space $S(n)$ if for every input x of length n, M uses at most $S(n)$ space.
- The input tape is **read-only** and does not count towards space usage.
- There is a designated **output tape** which is **write-only** and does not count towards space usage.
- Only the **work tapes** count towards space complexity.

Space Complexity Classes

Definition (Deterministic space classes)

- $\mathrm{DSPACE}[S(n)]$ is the class of languages decidable by a deterministic multi-tape Turing machine using $O(S(n))$ space.

Space Complexity Classes

Definition (Deterministic space classes)

- $\text{DSPACE}[S(n)]$ is the class of languages decidable by a deterministic multi-tape Turing machine using $O(S(n))$ space.
- $\text{TIMESPACE}[T(n), S(n)]$ (i.e., $\text{TISP}[T(n), S(n)]$) is the class of languages decidable by a deterministic multi-tape Turing machine using $O(T(n))$ time and $O(S(n))$ space.

Motivation: Hardness for NP, Continued

- "SAT $\in$ P?" is way too hard apparently (e.g., the relativization barrier), can we prove something weaker first?
- **A simpler question**: Is there a polynomial time algorithm for SAT that uses very little space? (say, is SAT $\in$ TISP$[n^{O(1)}, n^{0.01}]$?)
- **This lecture**: SAT is not in TISP$[n^{\phi-\epsilon}, n^{\epsilon}]$ for $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$ and very small $\epsilon > 0$!
- More precisely, we will prove the following:

Theorem

NTIME$[n] \not\subseteq$ TISP$[n^{\phi-\epsilon}, n^{\epsilon}]$ for $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$ and very small $\epsilon > 0$.

An alternative definition of NTIME$[T(n)]$

Instead of using a non-deterministic Turing machine, we can define NTIME$[T(n)]$ using a deterministic Turing machine with witness.

Definition (Alternative definition of NTIME[T(n))

] A language $L \in$ NTIME$[T(n)]$ if and only if there exists a deterministic Turing machine M and a constant c such that:

- For every $x \in L$, there exists a witness w with $|w| \leqslant c \cdot T(|x|)$ such that $M(x, w)$ accepts in time $c \cdot T(|x|)$.

An alternative definition of NTIME$[T(n)]$

Instead of using a non-deterministic Turing machine, we can define NTIME$[T(n)]$ using a deterministic Turing machine with witness.

Definition (Alternative definition of NTIME[T(n))

] A language $L \in \mathrm{NTIME}[T(n)]$ if and only if there exists a deterministic Turing machine M and a constant c such that:

- For every $x \in L$, there exists a witness w with $|w| \leqslant c \cdot T(|x|)$ such that $M(x, w)$ accepts in time $c \cdot T(|x|)$.
- For every $x \notin L$, for all strings w with $|w| \leqslant c \cdot T(|x|)$, $M(x, w)$ rejects in time $c \cdot T(|x|)$.

An alternative definition of NTIME$[T(n)]$

Instead of using a non-deterministic Turing machine, we can define NTIME$[T(n)]$ using a deterministic Turing machine with witness.

Definition (Alternative definition of NTIME[T(n))

] A language $L \in \mathrm{NTIME}[T(n)]$ if and only if there exists a deterministic Turing machine M and a constant c such that:

- For every $x \in L$, there exists a witness w with $|w| \leqslant c \cdot T(|x|)$ such that $M(x, w)$ accepts in time $c \cdot T(|x|)$.
- For every $x \notin L$, for all strings w with $|w| \leqslant c \cdot T(|x|)$, $M(x, w)$ rejects in time $c \cdot T(|x|)$.
- This is equivalent to the standard definition using non-deterministic Turing machines. The witness w corresponds to the sequence of non-deterministic choices.

Definition: $\Sigma_2\mathrm{TIME}[T(n)]$ and $\Pi_2\mathrm{TIME}[T(n)]$

Definition

A language $L \in \Sigma_2\mathrm{TIME}[T(n)]$ if and only if there exists a deterministic Turing machine M and a constant c such that:

- For every $x \in L$, there exists a witness w_1 with $|w_1| \leqslant c \cdot T(|x|)$ such that for all strings w_2 with $|w_2| \leqslant c \cdot T(|x|)$, $M(x, w_1, w_2)$ accepts in time $c \cdot T(|x|)$.
- For every $x \notin L$, for all strings w_1 with $|w_1| \leqslant c \cdot T(|x|)$, there exists a witness w_2 with $|w_2| \leqslant c \cdot T(|x|)$ such that $M(x, w_1, w_2)$ rejects in time $c \cdot T(|x|)$.

Definition: $\Sigma_2\text{TIME}[T(n)]$ and $\Pi_2\text{TIME}[T(n)]$

Definition

A language $L \in \Pi_2\text{TIME}[T(n)]$ if and only if there exists a deterministic Turing machine M and a constant c such that:

- For every $x \in L$, for all strings w_1 with $|w_1| \leqslant c \cdot T(|x|)$, there exists a witness w_2 with $|w_2| \leqslant c \cdot T(|x|)$ such that $M(x, w_1, w_2)$ accepts in time $c \cdot T(|x|)$.
- For every $x \notin L$, there exists a witness w_1 with $|w_1| \leqslant c \cdot T(|x|)$ such that for all strings w_2 with $|w_2| \leqslant c \cdot T(|x|)$, $M(x, w_1, w_2)$ rejects in time $c \cdot T(|x|)$.

The Speedup Lemma

$\mathrm{DTS}[n^c] = \mathrm{TISP}[n^c, n^{o(1)}]$.

Lemma (Speedup Lemma)

$DTS[n^d] \subseteq (\exists n^x)(\forall \log n)DTS[n^{d-x}]$.
$DTS[n^d] \subseteq (\forall n^x)(\exists \log n)DTS[n^{d-x}]$.

Definition

A language $L \in (\exists f(n))(\forall g(n))\mathrm{DTS}[n^k]$ if there is an $n^{o(1)}$ space machine M such that for all $x \in \{0,1\}^n$,

- $x \in L$ if and only if there exists a witness w with $|w| = f(n)^{1+o(1)}$ such that for every y with $|y| = g(n)^{1+o(1)}$, $M(x, w, y)$ accepts in $n^{k+o(1)}$ time.

Proof of the Speedup Lemma: see the white board!

The Slowdown Lemma

$\mathrm{DTS}[n^c] = \mathrm{TISP}[n^c, n^{o(1)}]$.

Lemma (Slowdown Lemma)

If $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$, then $\Sigma_2\mathsf{TIME}[n^d] \subseteq \mathsf{NTIME}[n^{d\cdot c+o(1)}]$.

Proof: see the white board!

The Padding Lemma

Lemma (Padding Lemma)

If $\mathit{NTIME}[n] \subseteq \mathit{DTS}[n^c]$, *then* $\mathit{NTIME}[n^d] \subseteq \mathit{DTS}[n^{d \cdot c}]$.

Proof: see the white board!

Warmup: The $\sqrt{2}$ Lower Bound

Theorem

$NTIME[n] \not\subseteq DTS[n^{\sqrt{2}-\epsilon}]$ *for any* $\epsilon > 0$.

Warmup: The $\sqrt{2}$ Lower Bound

Theorem

$NTIME[n] \not\subseteq DTS[n^{\sqrt{2}-\epsilon}]$ for any $\epsilon > 0$.

- Proof by contradiction.

Warmup: The $\sqrt{2}$ Lower Bound

Theorem

$NTIME[n] \not\subseteq DTS[n^{\sqrt{2}-\epsilon}]$ *for any* $\epsilon > 0$.

- Proof by contradiction.
- Assume $\mathrm{NTIME}[n] \subseteq \mathrm{DTS}[n^{\sqrt{2}-\epsilon}]$, we will deduce $\mathrm{NTIME}[n^2] \subseteq \mathrm{NTIME}[n^{2-\epsilon'}]$, $\epsilon' > 0$, contradiction to the NTIME hierarchy theorem!

Warmup: The $\sqrt{2}$ Lower Bound

Theorem

$NTIME[n] \not\subseteq DTS[n^{\sqrt{2}-\epsilon}]$ for any $\epsilon > 0$.

- Proof by contradiction.
- Assume $\mathrm{NTIME}[n] \subseteq \mathrm{DTS}[n^{\sqrt{2}-\epsilon}]$, we will deduce $\mathrm{NTIME}[n^2] \subseteq \mathrm{NTIME}[n^{2-\epsilon'}]$, $\epsilon' > 0$, contradiction to the NTIME hierarchy theorem!
- Proof: see the white board!

Main theorem: The ϕ Lower Bound

Theorem

$NTIME[n] \not\subseteq DTS[n^{\phi-\epsilon}]$ *for any* $\epsilon > 0$, $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$.

Lemma

$\Sigma_2 TIME[n^a] \not\subseteq \Pi_2 TIME[n^b]$ *for any* $a > b > 1$.

Main theorem: The ϕ Lower Bound

Theorem

$NTIME[n] \not\subseteq DTS[n^{\phi-\epsilon}]$ *for any* $\epsilon > 0$, $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$.

- Again, Proof by contradiction.

Lemma

$\Sigma_2 TIME[n^a] \not\subseteq \Pi_2 TIME[n^b]$ *for any* $a > b > 1$.

Main theorem: The ϕ Lower Bound

Theorem

$NTIME[n] \not\subseteq DTS[n^{\phi-\epsilon}]$ *for any* $\epsilon > 0$, $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$.

- Again, Proof by contradiction.
- Assume $\mathrm{NTIME}[n] \subseteq \mathrm{DTS}[n^{\phi-\epsilon}]$, we will deduce $\Sigma_2\mathrm{TIME}[n^a] \subseteq \Pi_2\mathrm{TIME}[n^b]$ for some $a > b > 1$, this is also a contradiction.

Lemma

$\Sigma_2 TIME[n^a] \not\subseteq \Pi_2 TIME[n^b]$ *for any* $a > b > 1$.

Main theorem: The ϕ Lower Bound

Theorem

NTIME$[n] \not\subseteq$ *DTS*$[n^{\phi-\epsilon}]$ *for any* $\epsilon > 0$, $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$.

Key lemma:

Lemma

For all $k \geqslant 0$, *if NTIME*$[n] \subseteq$ *DTS*$[n^c]$, *then*

$$DTS\left[n^{2+\sum_{i=1}^{k} c^i}\right] \subseteq \Sigma_2 TIME\left[n^{c^k+o(1)}\right]$$

and

$$DTS\left[n^{2+\sum_{i=1}^{k} c^i}\right] \subseteq \Pi_2 TIME\left[n^{c^k+o(1)}\right]$$

Proof: see the white board!