1 Time-Space Trade-off for SAT

Definition 1 For functions $T(n)$ and $S(n)$, $TISP(T(n), S(n))$ is the set of languages $L$ such that $\exists$ TM $M$ that decides $L$ in time $O(T(n))$ and space $O(S(n))$ on inputs of length $n$.

Remark Generally, $TISP(T(n), S(n)) \neq TIME(T(n)) \cap SPACE(S(n))$.

The main result of this lecture will be:

Theorem 2 $SAT \notin TISP(n^{1.1}, n^{0.1})$.

We will make use of the supporting theorems:

Theorem 3 $NTIME(n) \nsubseteq TISP(n^{1.2}, n^{0.2})$.

Proof Deferred to section 1.1.

Theorem 4 (Cook-Levin, strong) The decision problem for any $L \in NTIME(T(n))$ can be reduced, in time $O(T(n)\log(T(n)))$ and space $O(\log(T(n)))$, to the satisfiability of a $O(T(n)\log(T(n)))-$size formula.

Proof Seen in past lecture on the Cook-Levin theorem.

When we have established Theorem 3, we can give:

Proof of Theorem 2: Let $L \in NTIME(n)$. Reduce $x \in L$ in time $O(n\log(n))$ and space $O(\log(n))$ to $\varphi \in SAT$ of size $m = O(n\log(n))$. Suppose for a moment that $SAT \in TISP(n^{1.1}, n^{0.1})$ on instances of size $n$; then $\varphi$ is decidable in time $O(m^{1.1})$ and space $O(m^{0.1})$, therefore we can decide whether $\varphi$ is satisfiable in $TISP(n^{1.1}\log(n), n^{0.1}\log(n))$, and therefore $L \in TISP(n^{1.2}, n^{0.2})$, but this contradicts Theorem 3.

1.1 Proving Theorem 3

Establishing this will require a synthesis of results and concepts seen in lecture to date including the non-deterministic time hierarchy theorem, the polynomial hierarchy, and alternation. Our goal will be to find a way to trade space for alternations.

Lemma 5 The negation of Theorem 3 implies $NTIME(n^{10}) \subseteq TISP(n^{12}, n^{2})$.

Sketch of Proof By padding argument. On an input $x$ of length $n$, append $n^{10}$ zeroes to $x$ and call the result $x'$ so that $x'$ has length $m \approx n^{10}$.

Remark The converse does not hold.

Lemma 6 $TISP(n^{12}, n^{2}) \subseteq \Sigma_2$-TIME($n^{8}$).

Proof Deferred to section 1.1.1.

Lemma 7 The negation of Theorem 3 implies $\Sigma_2$-TIME($n^{8}$) $\subseteq NTIME(n^{9.6})$.

Proof Deferred to section 1.1.2.

When we have established Lemmas 6 and 14, we can give:

Proof of Theorem 3: Suppose the negation of Theorem 3. Then by lemmas 5, 6, and 14, we find $NTIME(n^{10}) \subseteq NTIME(n^{9.6})$, contradicting the non-deterministic time hierarchy theorem.
1.1.1 Proof of Lemma 6

We will get a handle on low-spaceness using configuration graphs.

Suppose \( L \in TISP(n^{12}, n^2) \). Then \( \exists TM \ M \) deciding \( L \) such that on input \( x \) of length \( n \), \( M \) takes time \( O(n^{12}) \) and space \( O(n^5) \).

Let \( G_{M,x} \) be the configuration graph of \( M \) on input \( x \). Then each configuration has length \( O(n^2) \) and \( M(x) = 1 \iff \exists \) a path from \( c_{\text{start}} \) to \( c_{\text{accept}} \) of length at most \( O(n^{12}) \).

**Proof Attempt** First attempt. Recall that in proving Savitch’s theorem we traded a path of length \( k \) for the existence of a midpoint connecting 2 paths of length \( k/2 \). That’s not quite enough here since we cannot allow arbitrarily many alternations while staying in \( \Sigma_2\text{-TIME} \).

**Proof Attempt** Second attempt. Formulate the problem as \( M(x) = 1 \iff \exists v_1, \ldots, v_n \) such that "for each" \( i \in [n^6] \) \( \exists \) a path from \( v_i \) to \( v_{i+1} \) of length \( n^6 \) and \( \exists \) paths from \( c_{\text{start}} \) to \( v_1 \) and from \( v_n \) to \( c_{\text{accept}} \) each of length \( n^6 \) where, here, by "for each", we mean that the overall predicate is (in part) a composition of separate predicates for each \( i \) in the range.

This is in \( \Sigma_1 = NP \). Each vertex label is of size \( O(n^2) \) and there are \( O(n^6) \) individual statements to write out. Therefore the overall statement is of length \( O(n^8) \). Each individual statement requires \( O(n^9) \) time to verify, so this proof attempt has succeeded only in demonstrating membership in \( NTIME(n^{12}) \), which is still not good enough.

**Proof** Let’s modify the second proof attempt to use the power of the \( \forall \) quantifier. We will have \( M(x) = 1 \iff \exists v_1, \ldots, v_{O(n^9)} \) such that \( v_1 = c_{\text{start}}, v_{O(n^9)} = c_{\text{accept}} \), and \( \forall i \in [n^6] \exists \) a path of length \( O(n^8) \) between \( v_i \) and \( v_{i+1} \). The portion of this predicate headed by the \( \forall \) quantifier requires time \( O(n^7) \) to verify, therefore the overall statement is in \( \Sigma_2\text{-TIME}(n^8) \).

1.1.2 Proof of Lemma 14

This one is a bit more straightforward. It will be similar to our proof that \( P = NP \) \( \Rightarrow \) the polynomial hierarchy collapses.

**Proof Idea** We’d like to somehow get rid of one of the quantifiers in a \( \Sigma_2\text{-TIME} \) statement. Suppose \( L \in \Sigma_2\text{-TIME}(n^8) \). Then \( \exists TM \ M \) such that \( \forall x \in \{0,1\}^n \) we have \( x \in L \iff \exists u \in \{0,1\}^{n^8} \) such that \( \forall v \in \{0,1\}^{n^8} \) we have \( M(x,u,v) = 1 \) where \( M \) runs in time \( O(n^8) \). If we can do the "\( \forall v \ldots \)" portion deterministically then we’ll have an \( NTIME \) statement. See that the negation of Theorem 3 \( \Rightarrow \) \( NTIME(n) \subseteq DTIME(n^{1.2}) \Rightarrow coNTIME(n) \subseteq DTIME(n^{1.2}) \).

**Proof** Let’s interpret \( (x,u) \) as the input for \( M \) rather than simply \( x \). Define \( L' \) by \( (x,u) \in L' \iff \forall v \in \{0,1\}^{n^8} M((x,u),v) = 1 \). Observe \( L' \in coNTIME(n) \) because \( |(x,u)| = |x| + |u| = n + n^8 = O(n^8) \). Then \( L' \in NTIME(n) \subseteq DTIME(n^{1.2}) \) again by the negation of Theorem 3, thus \( L' \in DTIME(n^{1.2}) \).

Therefore \( \exists \) DTM \( M' \) deciding \( L' \) in time \( O(n^{1.2}) \). Now see \( x \in L \iff \exists u \in \{0,1\}^{n^8} \) such that \( \forall v \in \{0,1\}^{n^8} M((x,u),v) = 1 \iff \exists u \in \{0,1\}^{n^8} \) such that \( M'(x,u) = 1 \). Finally, see \( M'(x,u) \) runs in time \( O(|x| + |u|^{1.2}) = O(|x|^{0.6}) = O(n^{9.6}) \), and therefore \( L' \) is in \( NTIME(n^{9.6}) \).

**Remark** It may be possible to sharpen the bounds of these lemmas by working out variations that reduce through a higher level of the polynomial hierarchy than \( \Sigma_2 \).
2 Boolean Circuits

Definition 8 (Boolean Circuits) For every \( n \in \mathbb{N} \), an \( n \)-input, single-output Boolean circuit is a directed acyclic graph with \( n \) sources (vertices with no incoming edges) and one sink (vertex with no outgoing edges). All nonsource vertices are called gates and are labeled with one of logical operations: \( \lor \), \( \land \) or \( \neg \). The vertices labeled with \( \lor \) and \( \land \) have fan-in equal to 2 and the vertices labeled with \( \neg \) have fan-in 1. The size of \( C \), denoted by \(|C|\), is the number of vertices in it.

If \( C \) is a Boolean circuit, and \( x \in \{0,1\}^n \) is some input, then the output of \( C \) on \( x \), denoted by \( C(x) \), is defined in the natural way. More formally, for every vertex \( v \) of \( C \), we give it a value \( \text{val}(v) \) as follows: \( \text{val}(v) = x_i \) if \( v \) is the \( i \)th input vertex then \( \text{val}(v) = x_i \) and otherwise \( \text{val}(v) \) is defined recursively by applying \( v \)'s logical operation on the values of its fan-in vertices. The output \( C(x) \) is the value of the output vertex.

Remark The above proof actually gives a stronger result than its statement: The circuit is not only of polynomial size but can also be computed in polynomial time. And the inclusion is proper, which will be proved by showing that there is an undecidable language in \( \mathcal{P} \text{-size Language Class} \) but not decidable.

Proof Let \( L \in \text{DTIME}(T(n)) \Rightarrow L \in \text{Size}(O(T(n)^2)) \).

Theorem 13 (Relation between \( \mathcal{P} \text{-size} \) and \( \mathcal{P} \)) \( \mathcal{P} \subseteq \mathcal{P} \text{-poly} \).

Remark The above proof actually gives a stronger result than its statement: The circuit is not only of polynomial size but can also be computed in polynomial time. And the inclusion is proper, which will be proved by showing that there is an undecidable language in \( \mathcal{P} \text{-poly} \) but not in \( \mathcal{P} \) (by definition).

Lemma 14 Let \( L \subseteq \{0,1\}^* \) be a unary language (i.e., \( L \subseteq 1^n : n \in \mathbb{N} \)). Then, \( L \in \mathcal{P} \text{-poly} \).

Definition 16 (Circuit Satisfiability) The language \( \text{CKT-SAT} \) is defined as \{\( C: C \) is a Boolean circuit such that \( \exists x \ C(x) = 1 \}\).
Theorem 17  CKT-SAT is NP-Complete.
Proof  Please refer to the proofs of Proposition 6 and Lemma 8 in the notes of lecture 2. ■

Theorem 18 (Karp-Lipton, KL80) If $NP \subseteq \text{P}^\text{poly}$, then $PH = \Sigma_2^p$.
Proof Idea  To show $PH = \Sigma_2^p$, it suffices to show that $\Pi_2^p \subseteq \Sigma_2^p$ and in particular it suffices to show that $\Sigma_2^p$ contains the $\Pi_2^p$-complete language $\Pi_2 SAT$. ■

Theorem 19 (Expressing a Boolean Function) Every function $f : \{0,1\}^n \rightarrow \{0,1\}$ can be computed by a Boolean circuit of size $O(2^n/n)$.

Theorem 20 (Existence of Hard Functions, Sha49a) For every $n > 1$, there exists a function $f : \{0,1\}^n \rightarrow \{0,1\}$ that cannot be computed by a circuit $C$ of size $2^n/(10n)$.
Proof  According to the structural properties of Boolean circuits, $\#\{C : |C| \leq S\} \leq 2^{3S \log S}$. Then $\#\{C : |C| \leq 2^n/(10n)\} \leq 2^{3\frac{2^n}{30n} \log \frac{2^n}{20n}} \leq 2^{3\frac{2^n}{30n}} \leq 2^{2^n/3} < 2^{2^n} = \#\{f : \{0,1\}^n \rightarrow \{0,1\}\}$. As the number of boolean functions with $n$ inputs is strictly larger than the number of boolean circuits of size $2^n/(10n)$, then there must exist some $n$-var function not computable by a $2^n/(10n)$-sized circuit. ■

Theorem 21 (Size Hierarchy Theorem) For every functions $T, T' : \mathbb{N} \rightarrow \mathbb{N}$ with $2^n/n > T'(n) > 10T(n) > n$, then $\text{Size}(T(n)) \subset \text{Size}(T'(n))$. 

4