Lecture 8

Time-Space Tradeoffs for SAT

April 22, 2004
Lecturer: Paul Beame
Notes: Geoff Mohr

Definition 8.1. $\mathsf{TIMESPACE}(T(n), S(n)) = \{L \in \{0,1\}^* \mid \exists$ offline, multitape TM M such that $L = L(M)$ and M uses time $O(T(n))$ and space $O(S(n))\ \}$.

One should note that, while $\mathsf{TIMESPACE}(T(n), S(n)) \subseteq \mathsf{TIME}(T(n)) \cap \mathsf{SPACE}(S(n))$, in general the definition is different from $\mathsf{TIME}(T(n)) \cap \mathsf{SPACE}(S(n))$, since a *single* machine is required to run in both time $O(T(n))$ and space $O(S(n))$.

We will prove a variant of the following theorem which is the strongest form of time-space tradeoffs currently known. The first result of this sort was shown by Fortnow; on the way to showing bounds like these we prove bounds for smaller running times due to Lipton and Viglas.

Theorem 8.1 (Fortnow-van Melkebeek). *Let $\phi = (\sqrt{5}+1)/2$. There is a function $s : \mathbb{R}^+ \to \mathbb{R}^+$ s.t. for all $c < \phi$, $\mathsf{SAT} \notin \mathsf{TIMESPACE}(n^c, n^{s(c)})$ and $\lim_{c\to 1} s(c) = 1$.*

We will prove a weaker form of this theorem that for $c < \phi$, $\mathsf{SAT} \notin \mathsf{TIMESPACE}(n^c, n^{o(1)})$. By the efficient reduction described in Lemma 5.4 of $\mathsf{NTIME}(T(n))$ to 3-SAT, which maps an $\mathsf{NTIME}(T(n))$ machine to a 3CNF formula of size $O(T(n)\log T(n))$ in time $O(T(n)\log T(n))$ time and $O(\log T(n))$ space, to prove this weaker form it suffices to show the following.

Theorem 8.2. *For $c < (\sqrt{5}+1)/2$, $\mathsf{NTIME}(n) \not\subseteq \mathsf{TIMESPACE}(n^c, n^{o(1)})$.*

One simple and old idea we will use is that of *padding* which shows that if simulations at low complexity levels exist then simulations at high complexity levels also exist.

Lemma 8.3 (Padding Meta-lemma). *Suppose that $\mathsf{C}(n) \subseteq \mathsf{C}'(f(n))$ for parameterized complexity class families C and C' and some function $f : \mathbb{N} \to \mathbb{R}^+$. C, C' parameterized by n and $f(n)$ respectively. Then for any $t(n) \geq n$ such that is computable using the resources allowed for $\mathsf{C}(t(n))$, $\mathsf{C}(t(n)) \subseteq \mathsf{C}'(f(t(n)))$.*

Proof. Let $L \in \mathsf{C}(t(n))$ and let M be a machine deciding L that witnesses this fact. Let $x \in \{0,1\}^n$ be input for M. Pad x to length $t(n)$, say to creat $x^{pad} = x01^{t(n)-n-1}$. For $A \in \mathsf{C}(t(n))$ let $A^{pad} = \{x01^{t(|x|)-|x|-1} \mid x \in A\ \}$. Since it is very easy to strip off the padding, $A^{pad} \in \mathsf{C}(n)$; so, by assumption, there is a witnessing machine M' showing that $A^{pad} \in \mathsf{C}'(f(n))$. We use M' as follows: On input x, create x^{pad} and run M' on x^{pad}. The resources used are those of $\mathsf{C}'(f(n))$ as required. □

Theorem 8.4 (Seiferas-Fischer-Meyer). *If $f, g : \mathbb{N} \to \mathbb{N}$, f, g are time constructible, and $f(n+1)$ is $o(g(n))$ then $\mathsf{NTIME}(f(n)) \subsetneq \mathsf{NTIME}(g(n))$.*

Note that, unlike the case for the deterministic time hierarchy, there is no multiplicative $O(\log T(n))$ factor in the separation. In this sense, it is stronger at low complexity levels. However for very quickly growing functions (for example double exponential), the condition involving the $+1$ is weaker than a multiplicative $\log T(n)$. When we use it we will need to ensure that the time bounds we consider are fairly small.

Proof Sketch. The general idea of the simulation begins in a similar way to the ordinary time hierarchy theorem: One adds a clock and uses a fixed simulation to create a universal Turing machine for all machines running in time bound $f(n)$. In this case, unlike th deterministic case we can simulate an arbitrary k-tape NTM by a 2-tape NTM with no more than a constant factor simulation loss. This allows one to add the clock with no complexity loss at all. The hard part is to execute the complementation part of the diagonalization argument and this is done by a sophisticated padding argument that causes the additional $+1$. It uses the fact that unary languages of arbitrarily high NTIME complexity exist. □

We will use the following outline in the proof of Theorem 8.2.

1. Assume $\mathsf{NTIME}(n) \subseteq \mathsf{TIMESPACE}(n^c, n^{o(1)})$ for some constant c.
2. Show that for some (not too large) $t(n)$ that is time computable, $\mathsf{NTIME}(n) \subseteq \mathsf{TIME}(n^c) \implies \mathsf{TIMESPACE}(t(n), t(n)^{o(1)}) \subseteq \mathsf{NTIME}(t(n)^{f(c)+o(1)})$ for some constant $f(c) > 0$.
3. For $T(n) = t(n)^{1/c}$ where $t(n)$ is given in part 2, put the two parts together derive

$$\begin{aligned} \mathsf{NTIME}(T(n)) &\subseteq \mathsf{TIMESPACE}(T(n)^c, T(n)^{o(1)}) && \text{from part 1 by padding Lemma 8.3} \\ &\subseteq \mathsf{NTIME}(T(n)^{cf(c)+o(1)}) && \text{by part 2.} \end{aligned}$$

4. If $cf(c) < 1$, this is a contradiction to the NTIME hierarchy given in Theorem 8.4.

Part 2 of this outline requires the most technical work. The key notions in the proof involve extensions of the ideas behind Savitch's Theorem and for this it is convenient to use alternating time complexity classes.

Definition 8.2. Define $\Sigma_k\mathsf{TIME}(T(n))$ to be the set of all languages L such that $x \in L$ iff $\exists^{T(|x|)} y_1 \forall^{T(|x|)} y_2 \ldots Q^{T(|x|)} y_k M(x, y_1, \ldots, y_k)$ where M runs for at most $O(T(|x|))$ steps on input $(x, y_1, \ldots, y_n)$. Define $\Pi_k\mathsf{TIME}(T(n))$ similarly.

Lemma 8.5. *For $S(n) \geq \log n$ and any integer function $b : \mathbb{N} \to \mathbb{N}$,*
$\mathsf{TIMESPACE}(T(n), S(n)) \subseteq \Sigma_2\mathsf{TIME}(T'(n))$ *where* $T'(n) = b(n) \cdot S(n) + T(n)/b(n) + \log b(n)$.

Proof. Recall from the proof of Savitch's Theorem that we can consider the computation as operating on the graph of TM configurations. For configurations C and C' we write $C \vdash^t C'$ if and only if configuration C yields C' in at most t steps. In the proof of Savitch's theorem we used the fact that we could assume a fixed form for the initial configuration C_0 and a unique accepting configuration C_f, and expressed

$$(C_0 \vdash^T C_f) \iff \exists C_m.\, ((C_0 \vdash^{T/2} C_m) \wedge (C_m \vdash^{T/2} C_f)).$$

In a similar way we can break up the computation into $b = b(n)$ pieces for any $b \geq 2$, so that, denoting C_f by C_b, we derive

$$(C_0 \vdash^T C_b) \iff \exists^{(b-1)S} C_1, C_2, \ldots, C_{b-1} \forall^{\log b} i.\, (C_{i-1} \vdash^{T/b} C_i).$$

Each configuration has size $O(S(n) + \log n) = O(S(n))$ and determining whether or not $C_{i-1} \vdash^{t/b} C_i$ requires time at most $O(T(n)/b(n) + S(n))$. Thus the computation overall is in $\Sigma_2\mathsf{TIME}(b(n) \cdot S(n) + T(n)/b(n) + \log b(n))$. □

Corollary 8.6. *For all* $T, S : \mathbb{N} \to \mathbb{R}^+$,
$\mathsf{TIMESPACE}(T(n), S(n)) \subseteq \Sigma_2\mathsf{TIME}(\sqrt{T(n)S(n)})$.

Proof. Apply Lemma 8.5 with $b(n) = \sqrt{\frac{T(n)}{S(n)}}$. □

Given a simple assumption about the simulation of nondeterministic time by deterministic time we see that we can remove an alternations from the computation.

Lemma 8.7. *If* $T(n) \geq n$ *is time constructible and* $\mathsf{NTIME}(n) \subseteq \mathsf{TIME}(T(n))$*, then for time constructible* $T'(n) \geq n, \Sigma_2\mathsf{TIME}(T'(n)) \subseteq \mathsf{NTIME}(T(T'(n)))$.

Proof. By definition, If $L \in \Sigma_2\mathsf{TIME}(T'(n))$ then there is some predicate R such that

$$x \in L \iff \exists^{T'(|x|)} y_1 \forall^{T'(|x|)} y_2 R(x, y_1, y_2)$$

and $R(x, y_1, y_2)$ is computable in time $O(T'(|x|))$. Therefore

$$x \in L \iff \exists^{T'(|x|)} y_1 \neg \exists^{T'(|x|)} y_2 \neg R(x, y_1, y_2).$$

By padding using the assumption that $\mathsf{NTIME}(n) \subseteq \mathsf{TIME}(T(n))$, we obtain $\mathsf{NTIME}(T'(n)) \subseteq \mathsf{TIME}(T(T'(n)))$ and thus the set

$$S = \{(x, y_1) \mid |y_1| \leq T'(|x|) \text{ and } \exists^{T'(|x|)} y_2 \neg R(x, y_1, y_2)\}$$

is in $\mathsf{TIME}(T(T'(n)))$. Since $x \in L$ if and only if $\exists^{T'(|x|)} y_1 \neg((x, y_1) \in S)$, it follows that

$$L \in \mathsf{NTIME}(T'(n) + T(T'(n))) = \mathsf{NTIME}(T(T'(n)))$$

as required. □

Note that the assumption in Lemma 8.7 can be weakened to $\mathsf{NTIME}(n) \subseteq \mathsf{coNTIME}(T(n))$ and the argument will still go through. We now obtain a simple version of Part 2 of our basic outline for the proof of Theorem 8.2.

Corollary 8.8. *If* $\mathsf{NTIME}(n) \subseteq \mathsf{TIME}(n^c)$ *then for* $t(n) \geq n^2$,
$\mathsf{TIMESPACE}(t(n), t(n)^{o(1)}) \subseteq \mathsf{NTIME}(t(n)^{c/2+o(1)})$.

Proof. By Corollary 8.6,

$$\mathsf{TIMESPACE}(t(n), t(n)^{o(1)}) \subseteq \Sigma_2\mathsf{TIME}(t(n)^{1/2+o(1)}).$$

Applying Lemma 8.7 with $T'(n) = t(n)^{1/2+o(1)} \geq n$ yields

$$\Sigma_2\mathsf{TIME}(t(n)^{1/2+o(1)}) \subseteq \mathsf{NTIME}(t(n)^{c/2+o(1)})$$

as required since $(t(n)^{o(1)})^c$ is $t(n)^{o(1)}$. □

Corollary 8.9 (Lipton-Viglas). $\mathsf{NTIME}(n) \not\subseteq \mathsf{TIMESPACE}(n^c, n^{o(1)})$ *for* $c < \sqrt{2}$.

Proof. Let $t(n)$ be as defined in Corollary 8.8 and let $T(n) = t(n)^{1/c}$. If $\mathsf{NTIME}(n) \subseteq \mathsf{TIMESPACE}(n^c, n^{o(1)})$ then

$$\begin{aligned}
\mathsf{NTIME}(T(n)) &\subseteq \mathsf{TIMESPACE}(T(n)^c, T(n)^{o(1)}) \\
&= \mathsf{TIMESPACE}(t(n), t(n)^{o(1)}) \\
&\subseteq \mathsf{NTIME}(t(n)^{c/2+o(1)}) \qquad \text{by Corollary 8.8} \\
&= \mathsf{NTIME}(T(n)^{c^2/2+o(1)}).
\end{aligned}$$

Since $c < \sqrt{2}$, $c^2/2 < 1$ and this yields a contradiction to the nondeterministic time hierarchy Theorem 8.4. □

Fortnow and van Melkebeek derived a slightly different form from Lemma 8.5 for alternating time simulation of $\mathsf{TIMESPACE}(T(n), S(n))$ using the following idea. For a deterministic Turing machine running for T steps we know that $C'_0 \vdash^T C_b$ if and only for all $C'_b \neq C_b$, $C'_0 \nvdash^T C'_b$. Furthermore Therefore

$$(C'_0 \vdash^T C_b) \iff \forall^{bS} C'_1, C'_2, \ldots, C'_{b-1}, C'_b \exists^{\log b} i.\ ((C'_b = C_b) \vee (C'_{i-1} \nvdash^{T/b} C'_i)).$$

Strictly speaking this construction would allow one to derive the following lemma.

Lemma 8.10. *For $S(n) \geq \log n$ and any integer function $b : \mathbb{N} \to \mathbb{N}$,*
$\mathsf{TIMESPACE}(T(n), S(n)) \subseteq \Pi_2\mathsf{TIME}(T'(n))$ *where* $T'(n) = b(n) \cdot S(n) + T(n)/b(n) + \log b(n)$.

This is no better than Lemma 8.5 but we will not use the idea in this simple form. The key is that we will be able to save because we have expressed $C'_0 \vdash^T C_b$ in terms of $C'_{i-1} \nvdash^{T/b} C'_i$. This will allow us to have fewer alternations when the construction is applied recursively since the $\neg\forall C'_1, \ldots$ quantifiers that will occur at the next level can be combined with the $\exists i$ quantifier at the current level. This is the idea behind the proof of Theorem 8.2.

Proof of Theorem 8.2. We first prove the following by induction on k. Let f_k be defined by $f_{k+1}(c) = c \cdot f_k(c)/(1 + f_k(c))$ and $f_1(c) = c/2$.

Claim 1. If $\mathsf{NTIME}(n) \subseteq \mathsf{TIME}(n^c)$ then for $k \geq 1$ and some not too large $t(n)$,

$$\mathsf{TIMESPACE}(t(n), t(n)^{o(1)}) \subseteq \mathsf{NTIME}(t(n)^{f_k(c)+o(1)}).$$

Proof of Claim. The base case $k = 1$ is precisely Corollary 8.8. Suppose that the claim is true for k. Suppose that we have a machine M witness a language L in $\mathsf{TIMESPACE}(t(n), t(n)^{o(1)})$. We apply the following expansion.

$$(C'_0 \vdash^t C_b) \iff \forall^{bS} C'_1, C'_2, \ldots, C'_{b-1}, C'_b \exists^{\log b} i.\ ((C'_b = C_b) \vee (C'_{i-1} \nvdash^{t/b} C'_i)).$$

Choose $b(n) = t(n)^{f_k(c)/(1+f_k(c)}$. Then $t(n)/b(n) = t(n)^{1/(f_k(c)+1)}$ and $b(n)s(n) = t(n)^{f_k(c)/(1+f_k(c))+o(1)}$. Since $f_k(c) \leq f_1(c) = c/2 < 1$, $b(n)s(n) \leq t(n)/b(n)$ so the computation time for the expansion is dominated by $t(n)/b(n) = t(n)^{1/(f_k(c)+1)}$. By the inductive hypothesis applied to the inner $C'_{i-1} \nvdash^{t/b} C'_i$ we obtain that for some not too large $t(n)/b(n)$ this computation can be done in

$$\mathsf{NTIME}([t(n)/b(n)]^{f_k(c)+o(1)}) = \mathsf{NTIME}(t(n)^{f_k(c)/(f_k(c)+1)+o(1)}).$$

Adding the $\exists^{\log b} i$ also keeps it in the same complexity class. By padding and the hypothesis that $\mathsf{NTIME}(n) \subseteq \mathsf{TIME}(n^c)$ we obtain that the inner computation $\exists^{\log b} i.\ ((C'_b = C_b) \vee (C'_{i-1} \not\vdash^{t/b} C'_i))$ can be done in

$$\mathsf{TIME}(t(n)^{c \cdot f_k(c)/(f_k(c)+1)+o(1)}).$$

Plugging this in we obtain that

$$\mathsf{TIMESPACE}(t(n), t(n)^{o(1)}) \subseteq \mathsf{coNTIME}(t(n)^{c \cdot f_k(c)/(f_k(c)+1)+o(1)}).$$

Since $\mathsf{TIMESPACE}(t(n), t(n)^{o(1)})$ is closed under complement and by the definition of $f_{k+1}(c)$ we obtain that

$$\mathsf{TIMESPACE}(t(n), t(n)^{o(1)}) \subseteq \mathsf{NTIME}(t(n)^{f_{k+1}(c)+o(1)})$$

as required. □

Applying the end of the proof outline we obtain that for any k, if $\mathsf{NTIME}(n) \subseteq \mathsf{TIMESPACE}(n^c, n^{o(1)})$ then for $T(n) = t(n)^{1/f_k(c)}$,

$$\begin{aligned}
\mathsf{NTIME}(T(n)) &\subseteq \mathsf{TIMESPACE}(T(n)^c, T(n)^{o(1)}) \\
&= \mathsf{TIMESPACE}(t(n), t(n)^{o(1)}) \\
&\subseteq \mathsf{NTIME}(t(n)^{f_k(c)+o(1)}) \\
&= \mathsf{NTIME}(T(n)^{c \cdot f_k(c)+o(1)}).
\end{aligned}$$

Observe that $f_k(c)$ is a monotonically decreasing function with fixed point $f_*(c) = c \cdot f_*(c)/(f_*(c)+1)$ when $f_*(c) = c-1$. Then $c \cdot f_*(c) = c(c-1) < 1$ when $c < \phi = (\sqrt{5}+1)/2$ which provides a contradiction to the nondeterministic time hierarchy theorem. □

Open Problem 8.1. Prove that $\mathsf{NTIME}(n) \subseteq \mathsf{TIMESPACE}(n^c, n^{o(1)})$ for some $c > (\sqrt{5}+1)/2$, for example $c = 2$.