Theorem (Savitch's theorem): for any function $s : \mathbb{N} \to \mathbb{N}$ s.t. $s = \Omega(\log)$,

$$\mathrm{NSPACE}(s) \subseteq \mathrm{SPACE}(s^2).$$

<u>Proof idea</u>: So far we have only talked about the configuration graphs of deterministic TMs. What about the configuration graphs of verifiers? (and for that matter of randomized algorithms)

Given a verifier M on input x, with certificate y and work tape w we define the configuration graph $G_{M,x}$ analogously to the deterministic TM case, that is, each vertex is given by a configuration of the TM M - given by tuple (q, h_i, h_w, w) where q is current state of M, $h_i \leftarrow$ position of head of input tape, $h_w \leftarrow$ position of head of work tape, $w \leftarrow$ contents of work tape.

Now, each vertex has at most two outgoing edges (one labeled 0 and the other labeled by 1) indicating the transition function given the current configuration and the current bit at the certificate tape y.

With this definition, note that $L \in \mathrm{NSPACE}(s)$ $\Leftrightarrow \exists$ TM M with $s_M = O(s)$ s.t.

"$x \in L \Leftrightarrow \exists y$ s.t. $M(x,y) = 1$"

i.e. there is a path in $G_{M,x}$ from the start configuration to some accepting configuration.

Note that there is a path iff there is a simple path, (a path of length $\leq$ # vertices $= 2^{O(s)}$), so we can bound y to say $y \in \{0,1\}^{2^{O(s)}}$.

We are now ready to prove Savitch's theorem.

<u>Proof</u>: $L \in \mathrm{NSPACE}(s) \Rightarrow \exists$ TM M with $s_M = O(s)$ s.t. $x \in L \Leftrightarrow \exists y \in \{0,1\}^{2^{O(s)}}$ s.t. $M(x,y) = 1$.

We can assume w.l.o.g. that $G_{M,x}$ has only one <u>accepting configuration</u> (by modifying M to erase the contents of its work tape and return both heads (input & work) to the initial position).

Hence, we have the following equivalence:

$\exists y \in \{0,1\}^{2^{O(s)}}$ s.t. $M(x,y) = 1 \Leftrightarrow$ there is a path of length $\leq 2^{O(s)}$ from the start configuration of $G_{M,x}$ to the accepting configuration of $G_{M,x}$.

Note that the above is an instance of the ST-CONN problem: $\langle G_{M,x}, \text{start config.}, \text{accepting config}, 2^{O(s)} \rangle$.

By last lecture, we can solve this problem in space $O(\log(2^{O(s)}) \cdot \log(|V(G_{M,x})|)) =$
$= O(s^2)$

$\therefore$ there is TM $N \in \mathrm{SPACE}(s^2)$ s.t.

$x \in L \Leftrightarrow N(x) = 1 \Rightarrow \mathrm{NSPACE}(s) \subseteq \mathrm{SPACE}(s^2)$ $\square$

Remark: note that we <u>cannot</u> write the graph $G_{M,x}$ explicitly, as that would take too much space. However, note that our algorithm for ST-CONN only needs a succinct description of the given graph (i.e., given two vertices, is there an edge between them) and we have such a description by the transition function of M on x.

Logarithmic space

Definition (log-space transducer): a log-space transducer is a TM with read-only input tape, a write-only output tape (i.e. writes on cell then moves right on that tape) and a read/write work tape.
The work tape may contain $O(\log n)$ cells for inputs of length n.
A log-space transducer computes a function $f: \Sigma^* \to \Sigma^*$ where $f(x)$ is the string remaining on the output tape after M halts when its input is x.
In this case, we call f a log-space computable function.

Definition (log-space reduction): language A is log-space reducible to language B, denoted by $A \leq_L B$ if there is $f: \Sigma^* \to \Sigma^*$ log-space computable s.t. $x \in A \iff f(x) \in B$.

Definition (NL-completeness): Language B is NL-complete if

- $B \in NL$
- $\forall A \in NL \quad A \leq_L B$.