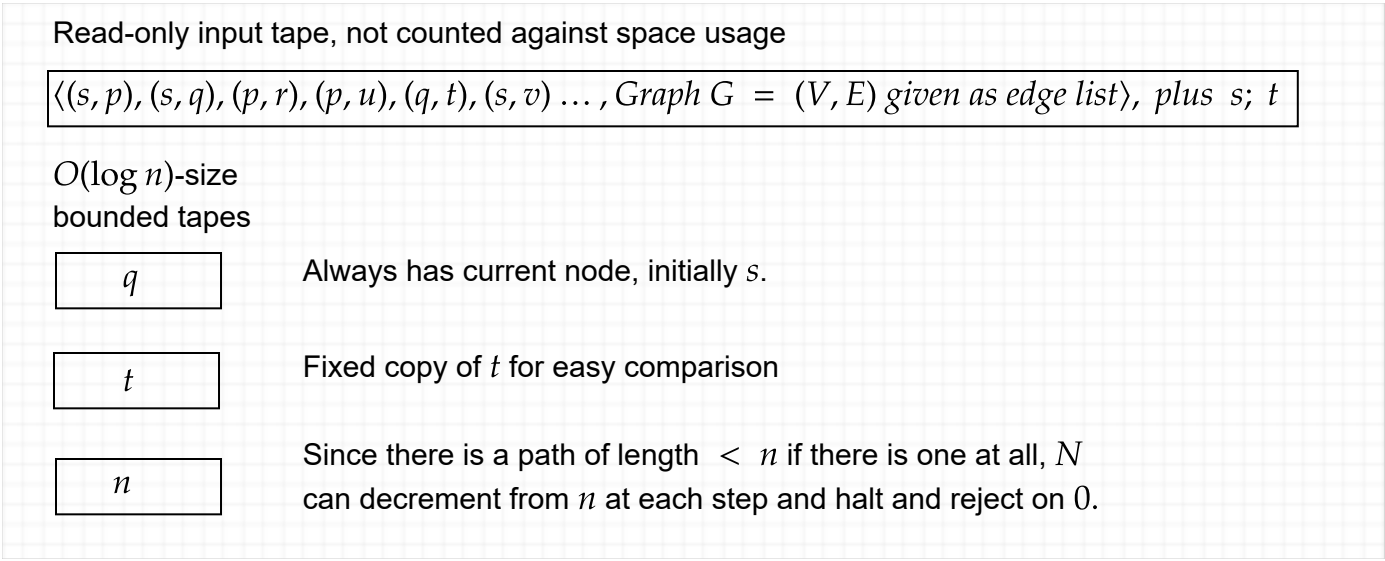
## CSE491/596 Lecture Mon. Oct. 30: Time and Space Complexity Classes

Say that a deterministic Turing machine $M$ **runs in space** $s(n)$ if for all $n$ and input strings $x$ of length $n$, $M(x)$ halts while **changing** at most $s(n)$ tape cells. For a nondeterministic TM $N$, we make the "conservative" definition that $N$ **runs in space** $s(n)$ if for all $n$ and input strings $x$ of length $n$, **all** computations by $N$ on input $x$ halt while changing at most $s(n)$ tape cells.

This definition allows a read-only input tape not to count against the space bound. By default, for these weeks, we will consider a multitape Turing machine to have a read-only **input tape** and any finite number of **worktapes**. Here again is the picture of the NTM for **GAP** from last Friday:




The worktape with $n$ prevents $N$ from getting into an infinite loop if it would keep following a cycle in the graph. Thus all computations by $N$ halt before trying more than $n$ steps in the trial path.

We make the following *simplifying assumption*: The space usage is either finite, i.e. $s(n) = O(1)$, or at least logarithmic, i.e., $s(n) = \Omega(\log n)$. [In fact, there are machines that use $\Theta(\log \log n)$ space in a meaningful manner, but they are esoteric and we will ignore them.]

**Definition**: DSPACE[$s(n)$] denotes the class of languages accepted by deterministic TMs that run in space $s(n)$. And NSPACE[$s(n)$] denotes the class of languages accepted by nondeterministic TMs that run in space $s(n)$. $O$-notation is understood collectively, i.e., DSPACE[$O(s(n))$] means the union of DSPACE[$cs(n) + d$] for all constants $c$ and $d$. Important cases of this definition have their own names:

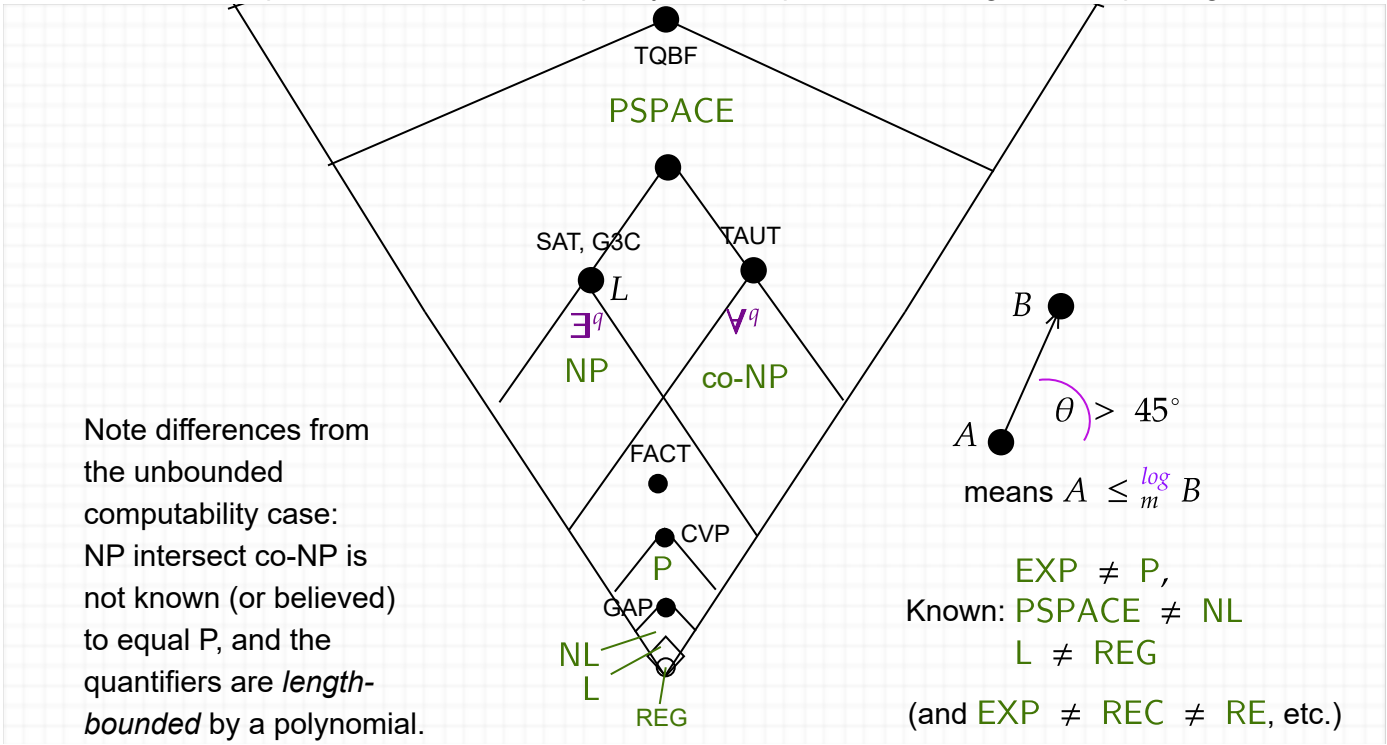
- DSPACE[$O(\log n)$] is called L for deterministic logarithmic space. When the L might be confused with a language, people write DLOG instead.
- NSPACE[$O(\log n)$] is called NL for nondeterministic logarithmic space. The NL generally doesn't get confused, but when people write DLOG for L, they write NLOG for NL.

- The class of *functions* computed in deterministic logarithmic space is sometimes denoted by FL (and not $FLOG$ which would be an ugly name). Likewise FP means functions computable in polynomial time. That FL $\subseteq$ FP is a case of a theorem to be proved below. We write $A \leq_m^{log} B$ to mean that $A$ mapping-reduces to $B$ via a function in FL. Log-space reducibility is thus a **refinement** of polynomial-time reducibility.
- DSPACE[$O(n)$] is called DLBA for (the class of languages decided by) deterministic linear bounded automata.
- NSPACE[$O(n)$] is called NLBA for (the class of languages decided by) nondeterministic linear bounded automata.
- DSPACE$\left[n^{O(1)}\right]$ is called PSPACE for deterministic polynomial space.
- NSPACE$\left[n^{O(1)}\right]$ would be called NPSPACE but we will see a theorem, called Savitch's Theorem, that makes this equal to PSPACE.

What about constant space? The following theorem is not as immediate as you might think:

**Theorem**: DSPACE[$O(1)$] = NSPACE[$O(1)$] = REG, meaning the class of regular languages.

If the Turing machines had to go left-to-right only on their read-only input tapes then the constant space usage would make them easy to convert into finite automata. But the Turing machines are allowed to move left on their input tapes. This complicates the proof. To show that the complication is inevitable: In terms of the constant space $k$ used by an NTM $N$, the smallest equivalent DFA can need a number of states that is **doubly exponential** in $k$. But it is still a DFA, regardless of the length $n$ of the input $x$, so the language $L(N)$ is regular. The rest of this lecture into Wednesday will establish the known inclusion relationships with time-based complexity classes per the following "landscape diagram":




Note differences from the unbounded computability case: NP intersect co-NP is not known (or believed) to equal P, and the quantifiers are *length-bounded* by a polynomial.

Under log-space reductions, because all languages in L are $\equiv_m^{\log}$ equivalent, the region for deterministic logspace should really warp into a point. If we could get a notion of "regular reductions" to suffice for all the NP-completeness reductions we want, we would only need to collapse REG into a point. As it stands, we can ignore little blemishes to get the big picture. It puts major classes for the four main complexity measures, viz. DTIME, DSPACE, NTIME, and NSPACE, onto one "landscape" map.

Before we go into weirdnesses like why NL shows as closed under complements whereas NP does not, we must establish the basic "positive knowledge" about which classes are included in others. We don't have much "negative knowledge" about non-inclusions between classes of different complexity measures at all. The central mystery is why this knowledge is so much less than what we know about separations between classes defined for the *same* complexity measure. The following theorem shows the yawning exponential gaps in our current best upper bounds.

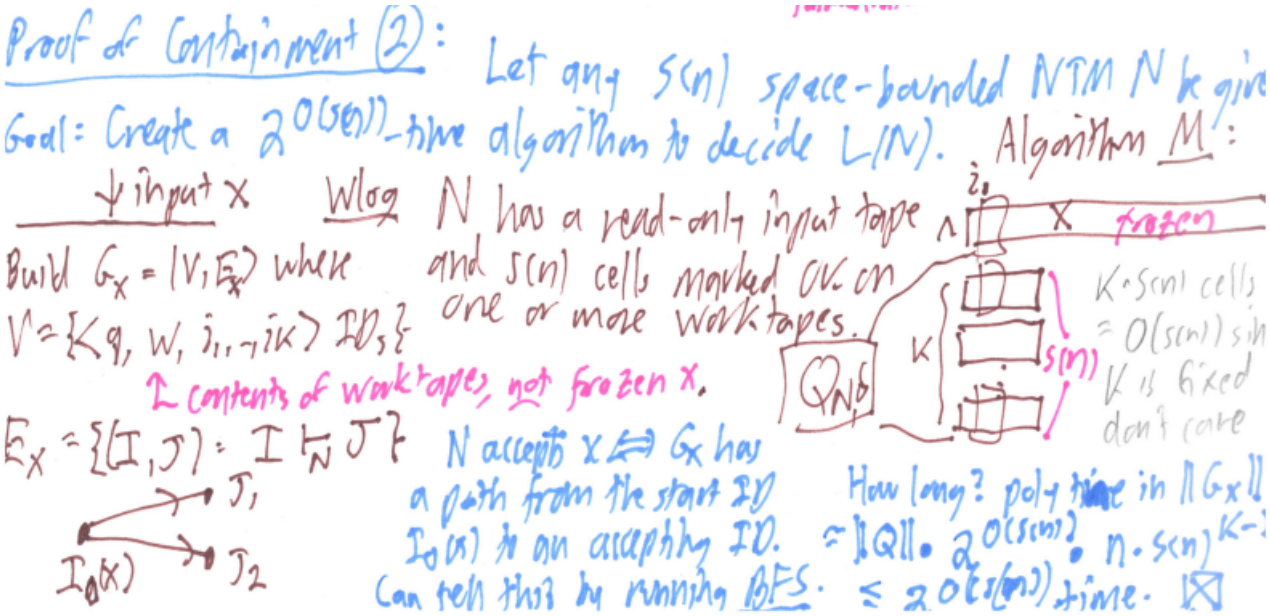
**Theorem**: For any "reasonable" time measure $t(n) \geq n+1$ and space measure $s(n) \geq \log_2 n$,

$$\text{DSPACE}[s(n)] \subseteq \text{NSPACE}[s(n)] \subseteq \text{DTIME}[2^{O(s(n))}]$$
$$\text{DTIME}[t(n)] \subseteq \text{NTIME}[t(n)] \subseteq \text{DSPACE}[O(t(n))] \subseteq \cdots$$

**Proof**: The first and third containments are immediate by definition. For the second, let $N$ be an NTM with some number $k$ of tapes and work alphabet $\Gamma$ that runs in space $s(n)$, and consider any input $x$ to $N$, putting $n = |x|$ as usual. The notion of "reasonable" allows us to lay out in advance $s(n)$ tape cells that $N$ is allowed to change. Thus any configuration $I$ has the form $I = \langle q, w, \overrightarrow{h} \rangle$ where $q$ is the current state, $w \in \Gamma^{s(n)}$ represents the current content of the cells $N$ can change, and $\vec{h}$ gives the head positions on all tapes, including the location of the input head reading $x$. Note that $I$ does not need to give the parts that don't change---if all cells occupied by $x$ are kept constant, $w$ doesn't need to include any of them. So the total number of different possible IDs we need to consider on input $x$ is at most

$$|Q| \cdot |\Gamma|^{s(n)} \cdot (n+2)(s(n) + 2k - 2)^{k-1}.$$

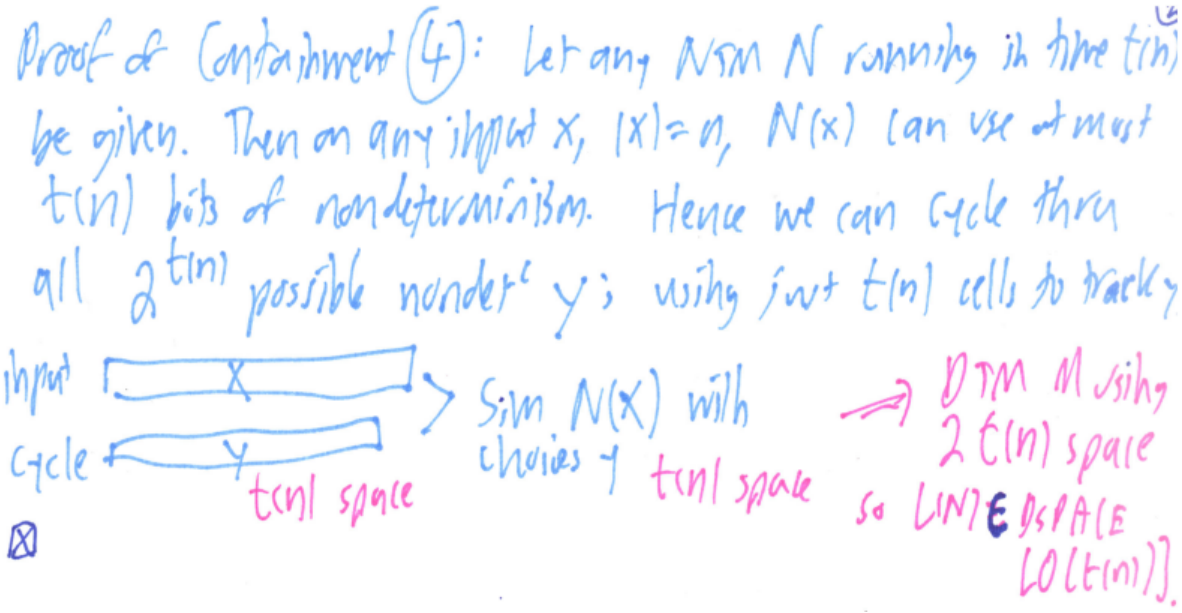
Since $s(n) \geq \log_2(n)$, $|\Gamma|^{s(n)}$ is at least $2^{\log_2(n)} = n$, so the third factor does not dominate the second factor and the whole size is bounded by $2^{O(s(n))}$. (The $+2$ and $2k-2$ allow the heads to occupy blanks to the left or right of $x$ and the cells they can change, however they are laid out on the tapes; they don't really matter to the $2^{O(s(n))}$ size estimate.)




Now we define a directed graph $G_x$ with the IDs $I, J, \ldots$ as its nodes and the relation $I \vdash_N J$ as its edge relation. Then $N$ accepts $x$ if and only if breadth-first search from the starting ID $I_0(x)$ finds an accepting ID (which by "good housekeeping" can be a unique node $t = I_f$). Since BFS runs in time polynomial in the size of the graph, and polynomial-in-$2^{O(s(n))}$ still gives $2^{O(s(n))}$, we obtain a deterministic algortihm that decides whether $x \in L(N)$ in time $2^{O(s(n))}$. This proves the second containment.

In-passing, we note that in the case $s(n) = O(\log n)$, $\mathsf{DTIME}\left[2^{O(s(n))}\right]$ is just P. Also, the mapping reduction $g_N(x) = \langle G_x, I_0(x), I_f \rangle$ is computable in logspace---because we can just treat the code of any ID $I$ as an $O(\log n)$-sized binary number and so lay out all the edges of $G_x$ using just the $\delta$ of the fixed NTM $N$ accepting a given language $A \in \mathsf{NL}$, we get that **GAP** is complete for NL under $\leq_m^{\log}$ reductions.

The fourth containment is (IMHO) best described as a *depth-first* search. Given a $k$-tape NTM $N$ that runs in time $t(n)$, we may suppose $N$ has binary nondeterminism, so that on any input $x$ of length $n$ there are at most $t(n)$ bits of nondeterminism that $N$ can use. We can organize all the possible guesses $y$ as branches of a binary tree $T$ of depth $t(n)$ and allocate $t(n)$ cells to hold the current $y$ we are trying. Since $N(x)$ cannot possibly use more than $kt(n)$ tape cells, we need only $t(n) + kt(n)$ space total to do a full transversal of $T$. We accept $x$ if and only if an accepting branch is found. This simulation takes roughly $2^{t(n)}$ time but it all operates within $O(t(n))$ space, so $L(N) \in \mathsf{DSPACE}[O(t(n))]$.




For example with $s(n) = O(\log n)$ we get

$$\mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE}.$$

This brings us back full-circle to the deterministic space measure, and we can ratchet up to the next level:

$$\mathsf{PSPACE} \subseteq \mathsf{NPSPACE} \subseteq \mathsf{EXP} \subseteq \mathsf{NEXP} \subseteq \mathsf{EXPSPACE}.$$

A reminder again that $\mathsf{EXP} = \mathsf{DTIME}[2^{n^{O(1)}}]$, not $\mathsf{DTIME}\left[2^{O(n)}\right]$, which is called $\mathsf{E}$. We do in fact have $\mathsf{PSPACE} = \mathsf{NPSPACE}$ by **Savitch's Theorem**, which we will prove next week in-tandem with showing that the language of true quantified Boolean formulas (called **TQBF** or, confusingly, **QBF**) is complete for $\mathsf{PSPACE}$ under $\leq_m^{\log}$ reductions. But basically the only multi-link chain of *differences* that we know from these classes is

$$\mathsf{NL} \subsetneq \mathsf{PSPACE} \subsetneq \mathsf{EXPSPACE}.$$

One can put $\mathsf{L}$ in place of $\mathsf{NL}$ here, and also prepend $\mathsf{REG}$ as a fourth proper link, but the main fact is the two exponential gaps thus-far seemingly needed to climb back around to the deterministic or nondeterministic space measure. Of final note today is the following theorem.

**Immerman-Szelépcsenyi Theorem**: For every space measure $s(n) = \Omega(\log n)$, $\mathsf{NSPACE}[s(n)]$ is closed under complements. In particular, $\mathsf{NL} = \text{co-}\mathsf{NL}$, and for linear space, $\mathsf{NLBA} = \text{co-}\mathsf{NLBA}$.

This was proved independently by Neil Immerman and Robert Szelépcsenyi in 1988. The proof is difficult and skipped here.

[The next lecture will cover the deterministic Time and Space Hierarchy Theorems in-tandem using a technical lemma from the notes https://cse.buffalo.edu/~regan/cse491596/CSE596inclusions.pdf ]