Computational Complexity Theory is the study of how much time and space it requires to solve computational problems. For any problem, a time function \( t(n) \) and a space function \( s(n) \), it has been an unexpected failure to find a solution between these bounds. For any function \( t(n) \), the corresponding function \( t(n) \) is defined.

\[
\begin{align*}
\text{DTIME}(t(n)) &= \{ M \mid M \text{ is a TM that for all } x, M(x) \text{ halts in } t(|x|) \text{ steps} \} \\
\text{NTIME}(t(n)) &= \{ L \mid L \text{ is an NFA that for all } x \text{ finds } \exists y \text{ s.t. } (x,y) \in L \text{ in } t(|x|) \text{ steps} \} \\
\text{DSPACE}(s(n)) &= \{ L \mid L \text{ is a TM that for all } x \text{ finds } \exists y \text{ s.t. } (x,y) \in L \text{ in } O(s(|x|)) \text{ steps} \} \\
\text{NSPACE}(s(n)) &= \{ L \mid L \text{ is an NFA that for all } x \text{ finds } \exists y \text{ s.t. } (x,y) \in L \text{ in } O(s(|x|)) \text{ steps} \}
\end{align*}
\]

It is OK to put \( O \)-notation inside the usual brackets, e.g., \( \text{DTIME}(\|n\|) \) or \( \text{DTIME}(n^2) \). For any class \( C \) of time (or space) functions, \( \text{DTIME}[C] = \bigcup_{c \in C} \text{DTIME}(c) \) and \( \text{NSPACE}[C] = \bigcup_{c \in C} \text{NSPACE}(c) \).

There are Hierarchies allowing us to remove an outcome (\( \cdots \)) but they are not naming or intent.

\[ \text{DTISP}(e(n), s(n)) = \{ (M,x) \mid M \text{ is a TM and } x \text{ halts in } e(n) \text{ steps} \} \]

The Most Basic Complexity Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>All problems solvable by a deterministic Turing machine in polynomial time</td>
</tr>
<tr>
<td>NP</td>
<td>A nondeterministic Turing machine in polynomial time solves these problems</td>
</tr>
<tr>
<td>PSPACE</td>
<td>The set of all problems solvable by a deterministic Turing machine in polynomial space</td>
</tr>
<tr>
<td>NPSPACE</td>
<td>The set of all problems solvable by a nondeterministic Turing machine in polynomial space</td>
</tr>
<tr>
<td>EXP</td>
<td>The set of all problems solvable by a deterministic Turing machine in exponential time</td>
</tr>
<tr>
<td>EXPSPACE</td>
<td>The set of all problems solvable by a deterministic Turing machine in exponential space</td>
</tr>
</tbody>
</table>

The complexity classes are related as follows:

\[ \text{P} \subseteq \text{NP} \subseteq \text{PSPACE} \subseteq \text{EXP} \]

In the hierarchy, \( \text{P} \) is the class of problems that can be solved in polynomial time by a deterministic Turing machine. \( \text{NP} \) is the class of problems that can be solved in polynomial time by a nondeterministic Turing machine. \( \text{PSPACE} \) is the class of problems that can be solved in polynomial space by a deterministic Turing machine. \( \text{EXP} \) is the class of problems that can be solved in exponential time by a deterministic Turing machine.
Complexity Theory is the study of how much time and memory are required to solve computational problems. For individual problems, it has been an unexpected failure of reasoning success.

The only separations known are:
- NL ⊆ ESPACE
- P ⊆ EXP
- NP ⊆ NEXP
- L ⊆ ESPACE follows, etc.
- NL is closed under ⊆

We can define det[b] logspace-comparable for 
A, B, C, and 

For Wednesday, please read 
§3 of AUR ch 27 on 
Boolean circuits, and 
§6 1-3 of AUR ch 28.

\[ L = \text{ESPACE}(0(n)) \]

\[ \text{NP} = \text{ESPACE}(0(n)) \]

There are classes \text{ESPACE}(0(n)) with 
...