Complexity Classes

L $= \text{DSPACE}(\log n)$

NL $= \text{NSPACE}(\log n)$

P $= \bigcup_{c>0} \text{DTIME}(n^c)$

NP $= \bigcup_{c>0} \text{NTIME}(n^c)$

PSPACE $= \bigcup_{c>0} \text{DPSACE}(n^c)$

NPSPACE $= \bigcup_{c>0} \text{NSPACE}(n^c)$

E $= \bigcup_{c>0} \text{DTIME}(2^{cn})$

NE $= \bigcup_{c>0} \text{NTIME}(2^{cn})$

EXP $= \bigcup_{c>0} \text{DTIME}(2^{n^c})$

NEXP $= \bigcup_{c>0} \text{NTIME}(2^{n^c})$

There're thousands more classes. But above are the most important and basic complexity classes.

Relationships

1. $\text{PSPACE} = \text{NPSPACE}$

   Proof. Using Savitch's Theorem.

2. $\text{NL} \subseteq \text{DSPACE}\ (\log^2 n)$

   Proof. Using Savitch's Theorem.

3. $\text{L} \subseteq \text{NL} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PSPACE} = \text{NSPACE} \subseteq \text{EXP} \subseteq \text{NEXP}$

4. $\text{P} \subsetneq \text{E} \subsetneq \text{EXP}$

   Proof. Time Hierarchy.

5. $\text{L} \subsetneq \text{PSPACE}$

   Proof. Space Hierarchy.

6. $\text{NP} \subsetneq \text{NE} \subsetneq \text{NEXP}$

   Proof. Time Hierarchy.

Now, the most important question in theoretical computer science is:

$$\text{P} \stackrel{?}{=} \text{NP}$$

NP Problems and NP-Completeness

Examples of NP problems are like composite number problem, perfect matching problem and clique problem. Clique has been proved to be an NP-Complete problem.

Definition 1. *A is polynomial time reducible to B, (denoted as $A \subseteq_m^P B$), if there is an $F \in FP$, s.t. $\forall x \in \Sigma^*$,*

$$x \in A \Leftrightarrow F(x) \in B$$

where $FP = \{f : \Sigma^* \rightarrow \Sigma^* \mid \exists$ a TM with output tape and on input x, $M(x)$ outputs $f(x)$, $M(x)$ runs in time $|x|^c$ for some constant $c.\}$.

Lemma 1. *If $A \subseteq_m^P$, then:*

1. *$B \in P \Rightarrow A \in P$.*
2. *$B \in NP \Rightarrow A \in NP$.*
3. *$A \notin P \Rightarrow B \notin P$.*
4. *$A \notin NP \Rightarrow B \notin NP$.*

Proof. We only sketch the proof of 1. The other three lemmas are similar.

Suppose $B \in$ P via TM M, and $A \subseteq_m^P B$ via function f. We need to construct a TM N that inputs on x, calculates $f(x)$ first and simulate M on $f(x)$. If f runs in time n^j, M runs in time n^k, then N requires

$$n^j + (n^j)^k = O(n^{jk})$$

in time. And since $A = L(M)$ apparently, we have $A \in$P. ◻

Definition 2. *Language L is said to be NP-Complete if*

1. *L$\in$ NP*
2. *$\forall A \in NP$, $A \subseteq_m^P L$*

An example of NP-Complete problem is

$$L = \{(<M>, x, 1^k) \mid M \text{ is NTM}, M(x) \text{ accepts in time } k\}$$

Below sketches the proof of its NP-Completeness:

Proof. We first prove that $L \in$ NP. To decide L, we need to simulate M for k steps. Since input size is larger than k, the time used by M will be at least $O(k^2)$. Hence $L \in$ NP.

Now assume $A \in$ NP and NTM N accepts A in time n^c. Define function $f(\cdot) = (<N>, x, 1^{|x|^c})$, we have

$$x \in A \Leftrightarrow N(x) \text{ accepts in time} |x|^c \Leftrightarrow (<N>, x, 1^{|x|^c}) \in L \Leftrightarrow f(x) \in L$$

therefore $A \subseteq_m^P L$. ◻

Lemma 2. *If A is NP-Complete, and*

1. *$B \in NP$*
2. *$A \subseteq_m^P B$*

then B is also NP-Complete.

Proof. Since A is NP-Complete, there exists $D \in$ NP s.t. $D \subseteq_m^P A$ via function f. Assume that $A \subseteq_m^P B$ via g, then:

$$x \in D \Leftrightarrow f(x) \in A \Leftrightarrow g(f(x)) \in B$$

Since f and g are both in polynomial time, $g(f(\cdot))$ is also in polynomial time. Therefore we have $D \subseteq_m^P B$. Because $B \in$ NP, we have that B is NP-Complete. □