

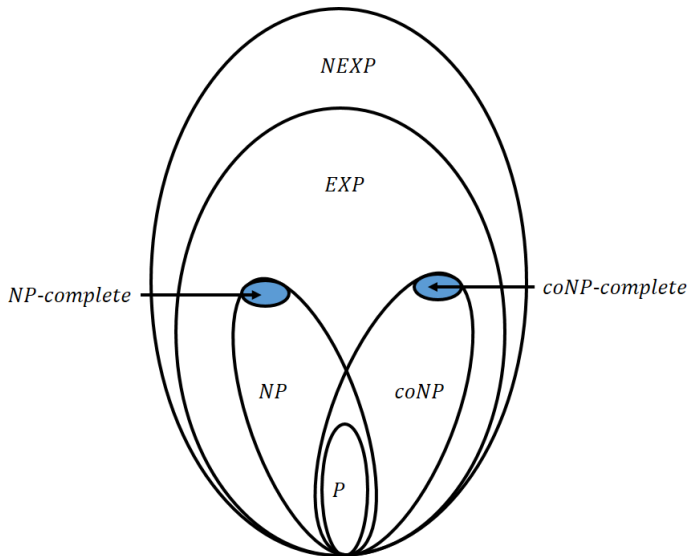
CSL853: Complexity Theory

Ragesh Jaiswal, CSE, IIT Delhi

Complexity Diagram

Complexity Diagram

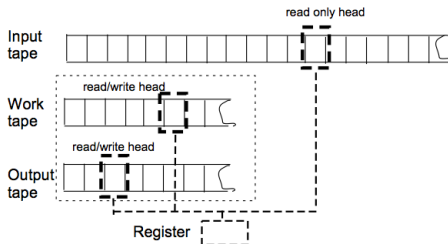
Our current view of complexity classes



Space Complexity

Space Complexity

SPACE and NSPACE



- For space bounded computation, only cells used in the read/write tape counts towards the space bound.
- We will focus on *space-constructible* functions $S : \mathbb{N} \rightarrow \mathbb{N}$.
 - There is a TM that computes $S(|x|)$ in $O(S(|x|))$ space given x as input.

Space Complexity

SPACE and NSPACE

Definition (SPACE and NSPACE)

Let $S : \mathbb{N} \rightarrow \mathbb{N}$ and $L \subseteq \{0, 1\}^*$. We say that $L \in \text{SPACE}(s(n))$ (resp. $L \in \text{NSPACE}(s(n))$) if there is a constant c and TM (resp. NDTM) M deciding L such that on every input $x \in \{0, 1\}^*$, the total number of locations that are at some point non-blank during M 's execution on x is at most $c \cdot s(|x|)$. (Non-blank locations in the read-only input tape do not count.)

Space Complexity

SPACE and NSPACE

Definition (SPACE and NSPACE)

Let $S : \mathbb{N} \rightarrow \mathbb{N}$ and $L \subseteq \{0, 1\}^*$. We say that $L \in \text{SPACE}(s(n))$ (resp. $L \in \text{NSPACE}(s(n))$) if there is a constant c and TM (resp. NDTM) M deciding L such that on every input $x \in \{0, 1\}^*$, the total number of locations that are at some point non-blank during M 's execution on x is at most $c \cdot s(|x|)$. (Non-blank locations in the read-only input tape do not count.)

- Show that: $\text{DTIME}(s(n)) \subseteq \text{SPACE}(s(n))$.

Theorem

For every space-constructible function $S : \mathbb{N} \rightarrow \mathbb{N}$,

$$\text{DTIME}(S(n)) \subseteq \text{SPACE}(S(n)) \subseteq \text{NSPACE}(S(n)) \subseteq \text{DTIME}(2^{O(S(n))})$$

Space Complexity

Space complexity classes

Definition (PSPACE and NPSPACE)

$$\begin{aligned}\text{PSPACE} &= \bigcup_{c>0} \text{SPACE}(n^c) \\ \text{NPSPACE} &= \bigcup_{c>0} \text{NPSPACE}(n^c)\end{aligned}$$

- Show that $3\text{SAT} \in \text{PSPACE}$.
- Show that $\text{NP} \subseteq \text{PSPACE}$.
- Show that if $\text{P} = \text{PSPACE}$, then $\text{P} = \text{NP}$.

Space Complexity

Space complexity classes

Definition (L and NL)

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

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

- Consider the following languages:
 - $\text{EVEN} = \{x : x \text{ has an even number of 1's}\}$
 - $\text{MULT} = \{(\langle n \rangle, \langle m \rangle, \langle nm \rangle) : n, m \in \mathbb{N}\}$
 - $\text{PATH} = \{(G, s, t) :$
 $G \text{ is a directed graph in which there is a path from } s \text{ to } t\}$
- Show that $\text{EVEN}, \text{MULT}, \text{PATH} \in \text{NL}$.

Space Complexity

Space hierarchy theorems

Theorem (Space Hierarchy Theorem, Stearns, Hartmanis, Lewis, 1965)

If f and g are space constructive functions satisfying $f(n) = o(g(n))$, then

$$\text{SPACE}(f(n)) \subsetneq \text{SPACE}(g(n)).$$

Definition (PSPACE-complete)

A language A is PSPACE-hard if for every $L \in \text{PSPACE}$, $L \leq_p A$. If in addition $A \in \text{PSPACE}$, then A is PSPACE-complete.

- Is there a language that is PSPACE-complete?

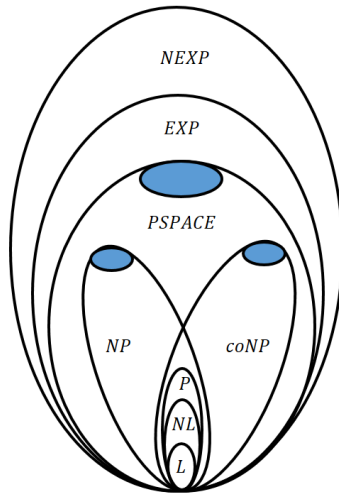
Definition (PSPACE-complete)

A language A is PSPACE-hard if for every $L \in \text{PSPACE}$, $L \leq_p A$. If in addition $A \in \text{PSPACE}$, then A is PSPACE-complete.

- Is there a language that is PSPACE-complete?
 - $L = \{\langle M, w, 1^n \rangle : \text{DTM } M \text{ accepts } w \text{ in space } n\}$.
- Is there a more natural problem that is PSPACE-complete?

Complexity Diagram

Our current view of complexity classes



Space Complexity

PSPACE-completeness

Definition (Quantified Boolean Formula (QBF))

A quantified boolean formula is a boolean formula of the form

$$Q_1x_1 Q_2x_2 \dots Q_nx_n \ \psi(x_1, x_2, \dots, x_n)$$

where each Q_i is one of the two quantifiers $\forall$ or $\exists$,
 $x_1, \dots, x_n \in \{0, 1\}$, and ψ is an unquantified boolean formula.

Definition (TQBF)

TQBF is the set of all quantified boolean formulas that are true.

Theorem (Stockmeyer and Meyer, 1973)

TQBF is PSPACE-complete.

Theorem (Stockmeyer and Meyer, 1973)

TQBF is PSPACE-complete.

Proof.

- Claim 1: $\text{TQBF} \in \text{PSPACE}$.
 - Claim 1.1: Let ψ be a QBF of size m with n variables, then the truth value of ψ can be computed in $O(n \cdot m)$ space.

Space Complexity

PSPACE-completeness

Theorem (Stockmeyer and Meyer, 1973)

TQBF is PSPACE-complete.

Proof.

- Claim 1: $\text{TQBF} \in \text{PSPACE}$.
- Claim 2: For every $L \in \text{PSPACE}$, $L \leq_p \text{TQBF}$.
 - Let M be the machine that decides L in $S(n)$ space.
 - Let $m = O(S(n))$ be the number of bits needed to encode the configurations of M on length n inputs.
 - Claim 2.1: There is a boolean formula $\psi_{M,x}$ such that for every pair of configurations (C, C') , $\psi_{M,x}(C, C')$ is true iff C and C' are adjacent configurations.

Theorem (Stockmeyer and Meyer, 1973)

TQBF is PSPACE-complete.

Proof.

- Claim 1: $\text{TQBF} \in \text{PSPACE}$.
- Claim 2: For every $L \in \text{PSPACE}$, $L \leq_p \text{TQBF}$.
 - Let M be the machine that decides L in $S(n)$ space.
 - Let $m = O(S(n))$ be the number of bits needed to encode the configurations of M on length n inputs.
 - Claim 2.1: There is a boolean formula $\psi_{M,x}$ such that for every pair of configurations (C, C') , $\psi_{M,x}(C, C')$ is true iff C and C' are adjacent configurations.
 - $\psi = \psi_m(C_{\text{start}}, C_{\text{accept}})$, $\psi_0(C, C') = \psi_{M,x}(C, C')$, and

$$\psi_i(C, C') = \exists C'', \psi_{i-1}(C, C'') \wedge \psi_{i-1}(C'', C')$$

- What is the size of the resulting formula?

Space Complexity

PSPACE-completeness

Theorem (Stockmeyer and Meyer, 1973)

TQBF is PSPACE-complete.

Proof.

- Claim 1: $\text{TQBF} \in \text{PSPACE}$.
- Claim 2: For every $L \in \text{PSPACE}$, $L \leq_p \text{TQBF}$.
 - Let M be the machine that decides L in $S(n)$ space.
 - Let $m = O(S(n))$ be the number of bits needed to encode the configurations of M on length n inputs.
 - Claim 2.1: There is a boolean formula $\psi_{M,x}$ such that for every pair of configurations (C, C') , $\psi_{M,x}(C, C')$ is true iff C and C' are adjacent configurations.
 - $\psi = \psi_m(C_{\text{start}}, C_{\text{accept}})$, $\psi_0(C, C') = \psi_{M,x}(C, C')$, and

$$\begin{aligned}\psi_i(C, C') &= \exists C'' \forall P \forall Q ((P = C \wedge Q = C'') \vee (P = C'' \wedge Q = C')) \\ &\Rightarrow \psi_{i-1}(P, Q)\end{aligned}$$

- What is the size of the resulting formula?

Theorem (Savitch's Theorem, Savitch, 1970)

For any space constructible $S : \mathbb{N} \rightarrow \mathbb{N}$ with $S(n) \geq \log n$, $\text{NSPACE}(S(n)) \subseteq \text{SPACE}(S(n)^2)$.

Proof.

- Let $L \in \text{NSPACE}(S(n))$ be decided by a TM M in space $S(n)$.
- Let the configuration graph $G = G_{M,x}$ contain $m = 2^{O(S(n))}$ vertices.
- We can decide if $x \in L$ if we can compute the following function: $\text{REACH}(u, v, i) = 1$ iff v is reachable from u in at most 2^i steps.

Theorem (Savitch's Theorem, Savitch, 1970)

For any space constructible $S : \mathbb{N} \rightarrow \mathbb{N}$ with $S(n) \geq \log n$, $\text{NSPACE}(S(n)) \subseteq \text{SPACE}(S(n)^2)$.

Proof.

- Let $L \in \text{NSPACE}(S(n))$ be decided by a TM M in space $S(n)$.
- Let the configuration graph $G = G_{M,x}$ contain $m = 2^{O(S(n))}$ vertices.
- We can decide if $x \in L$ if we can compute the following function: $\text{REACH}(u, v, i) = 1$ iff v is reachable from u in G in at most 2^i steps.
- Claim: The above function can be computed in $O(\log^2 m) = O(S(n)^2)$ space.



End