CS310 Automata Theory – 2017-2018

Nutan Limaye

Indian Institute of Technology, Bombay
nutan@cse.iitb.ac.in

Module 4: Effective computation

Module IV: Effective computation

Turing machines with resource constraints.

- Resources for computation.
  - Time: the number steps for which the TM runs
  - Space: the number of different cells on which the TM writes
  - The number of times an input bit can be read
  - The amount of energy used
  - ⋮
- Why bound resorces?
  - Viewing TM as algorithms.
  - TM to help in computation of important problems.
  - Finer study of decidable languages.
- How should we bound the resources?
  - Many different ways exist. ...

Time complexity and complexity classes

Let $t : \mathbb{N} \to \mathbb{N}$.

Definition

A language $L \subseteq \Sigma^*$ is said to be in class TIME$(t(n))$ if there exists a deterministic Turing machine M such that $\forall x \in \Sigma^*$,

- M halts on x in time $O(t(|x|))$, where $|x|$ indicates the length of x.
- if $x \in L$ then M accepts x.
- if $x \notin L$ then M rejects x.

$$P = \bigcup_k \text{TIME}(n^k)$$

$$EXP = \bigcup_k \text{TIME}(2^{n^k})$$

Time complexity and complexity classes

Let $t : \mathbb{N} \to \mathbb{N}$.

Definition

A language $L \subseteq \Sigma^*$ is said to be in class $\text{NTIME}(t(n))$ if there exists a non-deterministic Turing machine M such that $\forall x \in \Sigma^*$,

- each run of M halts on x in time $O(t(|x|))$, where $|x|$ indicates the length of x.
- if $x \in L$ then M accepts x on at least one run.
- if $x \notin L$ then M rejects x on all runs.

$$NP = \bigcup_k \text{NTIME}(n^k)$$

$$NEXP = \bigcup_k \text{NTIME}(2^{n^k})$$

Relationships between models

Lemma

Let $t(n) > n$. Let L be a language decided by a multitape TM in time $t(n)$. Then there is a single tape TM that decides L in time $O((t(n))^2)$.

Proof idea:

East step of multitape machine can be executed on a single tape machine in time $O(t(n))$.

Lemma

Let $t(n) > n$. Let L be a language decided by a non-deterministic TM in time $t(n)$. Then there is a deterministic TM that decides L in time $2^{O(t(n))}$.

Proof idea: DFS or BFS.

Relationships between complexity classes

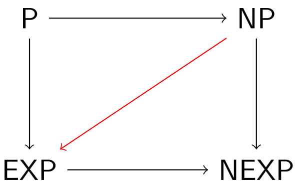
How are P, NP, EXP, and NEXP related?

P $\subseteq$ NP by definition.

P $\subseteq$ EXP again by definition.

Similarly, NP $\subseteq$ NEXP by definition.

Finally, NP $\subseteq$ EXP due to the previous lemma.



P vs. NP

P	the class of languages where membership can be decided quickly.
NP	the class of languages where membership can be verified quickly.

Examples

SAT $= \{\phi \mid \phi$ is satisfiable$\}$. in NP (and not known to be in P)

Reach $= \{(G, s, t) \mid t$ is reachable from s in $G\}$. in P

3-SAT $= \{\phi \mid \phi$ is a 3-CNF and satisfiable$\}$. in NP (and not known to be in P)

2-SAT $= \{\phi \mid \phi$ is a 2-CNF and satisfiable$\}$. in P

Factoring $= \{(k, n) \mid n$ has a factor $\leq k\}$. Google it!

Clique $= \{(G, k) \mid G$ has a clique of size $\geq k\}$. in NP (and not known to be in P)

Time heirarchy theorem

How do we separate NP from P?

To prove	Method used
not regular	pumping lemma for REG
non-context-free	pumping lemma or CFLs
not recognizable	diagonalization
not decidable	Rice's theorem or diagonalization and reductior
not in P	???

Finer structure inside P

Definition

A function $t : \mathbb{N} \to \mathbb{N}$ is said to be time constructible if the there exists a TM that on input 1^n, it outputs $t(n)$ in time $O(t(n))$.

Examples

n^2, $n \log n$.

Theorem

Let $t : \mathbb{N} \to \mathbb{N}$ be a time constructible function. There exists a language L such that $L \in TIME(t(n)^2)$, but $L \notin TIME(o(t(n)))$.

Polynomial time reductions and NP-hardness

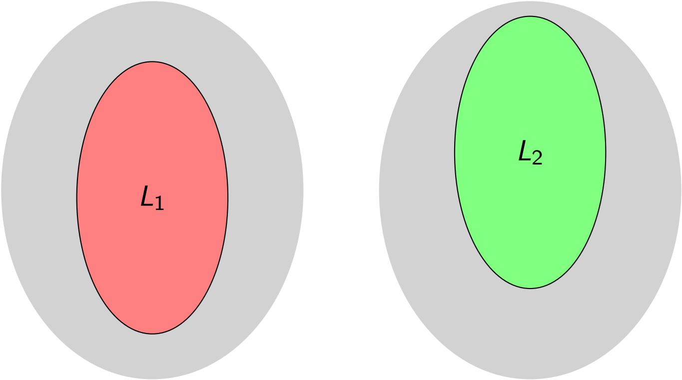
Definition

A function $f : \Sigma^* \rightarrow \Sigma^*$ is polynomial time computable if there is a polynomial time Turing machine TM, say M, such that on any input $w \in \Sigma^*$, M stops with only $f(w)$ on its tape.

Polynomial time reductions and NP-hardness

Definition

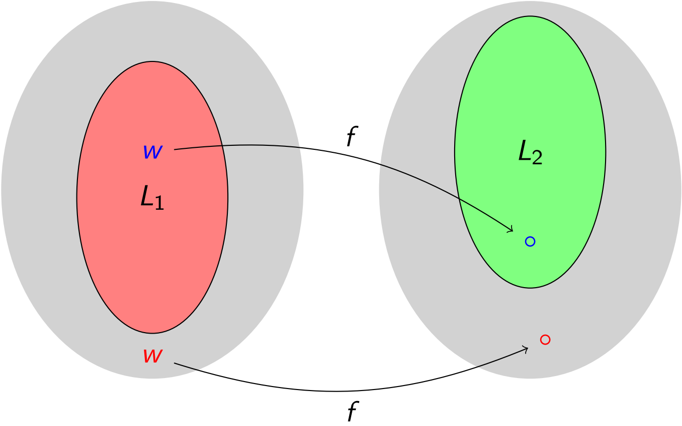
A language L_1 is said to be polynomial time reducible to another language L_2, denoted as $L_1 \leq_m L_2$, if there exists a polynomial time computable function f such that for all $w \in \Sigma^*$, $w \in L_1 \Leftrightarrow f(w) \in L_2$.



Polynomial time reductions and NP-hardness

Definition

A language L_1 is said to be polynomial time reducible to another language L_2, denoted as $L_1 \leq_m L_2$, if there exists a polynomial time computable function f such that for all $w \in \Sigma^*$, $w \in L_1 \Leftrightarrow f(w) \in L_2$.



Polynomial time reductions and NP-hardness

Definition

A language L is said to be NP-hard if for every language $L' \in \text{NP}$, there is a polynomial time reduction such that $L' \leq_m L$.

Definition

A language L is said to be NP-complete if the following two conditions hold:

- L is in NP.
- L is NP-hard.

Theorem ([Cook-Levin, 1970])

SAT is NP-complete.

Space bounded Turing Machines

The Turing Machine model with space bounds

- The input tape is assumed to be read-only.
- The space required to write down the input is not counted towards the space of the machine.
- The output tape assumed to be write-only.
- The space required to write down the output is not counted towards the space of the machine.

Space complexity and complexity classes

Let $s : \mathbb{N} \to \mathbb{N}$.

Definition

A language $L \subseteq \Sigma^*$ is said to be in class SPACE($s(n)$) if there exists a deterministic Turing machine M such that $\forall x \in \Sigma^*$,

- M halts on x using at most space $O(s(|x|))$,
- where $|x|$ indicates the length of x.
- if $x \in L$ then M accepts x.
- if $x \notin L$ then M rejects x.

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

$$PSPACE = \bigcup_k \text{SPACE}(n^k)$$

Examples of languages in Log

Min = $\{(w_1, w_2, \ldots, w_n, i) \mid w_i$ is the minimum among $w_1 \ldots w_n\}$.

Deg = $\{(G = (V, E), d, i) \mid v_i$ has degree $d\}$.

ADD = $\{(u, v, i) \mid i$th bit of $u + v$ is 1$\}$.

Verify-SAT

$= \{(\phi, a) \mid a = a_1, a_2, \ldots, a_n$ is an assignment satisfying $\phi\}$.

Space complexity and complexity classes

Let $s : \mathbb{N} \to \mathbb{N}$.

Definition

A language $L \subseteq \Sigma^*$ is said to be in class $\mathrm{NSPACE}(s(n))$ if there exists a non-deterministic Turing machine M such that $\forall x \in \Sigma^*$,

- M halts on x using at most space $O(s(|x|))$ on any run of the machine,
- where $|x|$ indicates the length of x.
- if $x \in L$ then there exists an accepting run of M on x.
- if $x \notin L$ then M rejects x on all the runs.

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

$$\mathrm{NPSPACE} = \bigcup_k \mathrm{NSPACE}(n^k)$$

Example of a language in NL

$$\text{Reach} = \{(G = (V, E), s, t) \mid \text{there is a path in } G \text{ from } s \text{ to } t\}$$

```
current ← s; count ← 0;
while count < n + 1 or current ≠ t;
{
    next ← non-det. guess a vertex from neighbors of current;
    current ← next;
    count ++;
}
if current = t then accept;
else reject;
```

NL is contained in P

Configurations of a non-deterministic space bounded machine.

Configuration of a space bounded Turing machine M

index: input head position (uses $O(\log n)$ bits)

data: the working space bits (uses $O(s(n))$ bits)

S_M: machine related information (Q, δ) (uses $O(1)$ bits)

A typical configuration $\langle \text{index, data}, S_M \rangle$

Let $\mathcal{C}_M$ be the set of all possible configuration of M.

Let C_0 be the initial configuration.

Let C_{acc} be the accepting configuration.

NL is contained in P

Definition

Let L be a language in $NSPACE$(s(n)) with TM M. Let C, C' be two configurations in $\mathcal{C}_M$. We say that a configuration C yields C' on input w if the machine M in one step goes from C to C' on input w.

Configurations Graph of M on input w.

Let $\mathcal{E}_{M,w} = \{(C, C') \mid C, C' \in \mathcal{C}_{\mathcal{M}}$ and C yields C' on input $w\}$

Let $\mathcal{G}_{M,w} = (\mathcal{C}_M, \mathcal{E}_{M,w})$

Let $\mathcal{G}_{M,w}$ be the configuration graph of M on w.

NL is contained in P

Theorem

If L is in $NSPACE(s(n))$ then L is in $TIME(2^{O(s(n))})$.

We know that $L \in NSPACE(\mathrm{s(n)})$. Let M be the machine.

First note that, $w \in L$ if and only if C_{acc} is reachable from C_0 in $\mathcal{G}_{M,w}$.

On any input w, the graph $\mathcal{G}_{M,w}$ can be computed in time $TIME(2^{O(s(n))})$.

- $|\mathcal{C}_M| = 2^{O(s(n))}$.
- Given C, C', checking whether $(C, C') \in \mathcal{E}_{M,w}$ or not is checkable in time $2^{O(s(n))}$.

Checking whether C_{acc} is reachable from C_0 can be checked in time $2^{O(s(n))}$.

- Reachability in a graph of size $2^{O(s(n))}$.

Corollary

NL is contained in P.