Lecture 14: Savitch's Theorem

Date: October 10, 2023.

Time and Space Bounds: Let $T : \mathbb{N} \to \mathbb{N}$ and $S : \mathbb{N} \to \mathbb{N}$ be functions.

- A (deterministic/nondeterministic) Turing machine M is said to **run in time** $T(n)$ (or is $T(n)$ **time bounded**) if on all inputs x, **all** computations of M on x take at most $T(|x|)$ steps before halting.
- A(deterministic/nondeterministic) Turingmachine$M$ is said to **runs in space** $S(n)$ (or is $S(n)$ **space bounded**) if on all inputs x, **all** computations of M on x use at most $S(|x|)$ worktape cells.

Time/Space Classes:

$$\begin{aligned}
\text{DTIME}(T(n)) &= \{\mathbf{L}(M) \mid M \text{ is a DTM running in time } T(n)\}\\
\text{NTIME}(T(n)) &= \{\mathbf{L}(M) \mid M \text{ is a NTM running in time } T(n)\}\\
\text{DSPACE}(S(n)) &= \{\mathbf{L}(M) \mid M \text{ is a DTM running in space } S(n)\}\\
\text{NSPACE}(S(n)) &= \{\mathbf{L}(M) \mid M \text{ is a NTM running in space } S(n)\}
\end{aligned}$$

If $\mathcal{A}$ is a collection of problems then co-$\mathcal{A} = \{L \mid \overline{L} \in \mathcal{A}\}$.

Theorem 1 (Linear Speedup/Compression). *For $T(n) \geq n+1$ and any constant $c \geq 1$*

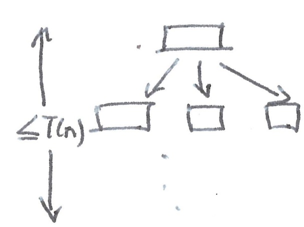
$$\begin{aligned}
\text{DTIME}(cT(n)) &\subseteq \text{DTIME}(T(n))\\
\text{NTIME}(cT(n)) &\subseteq \text{NTIME}(T(n))\\
\text{DSPACE}(cS(n)) &\subseteq \text{DSPACE}(S(n))\\
\text{NSPACE}(cS(n)) &\subseteq \text{NSPACE}(S(n))
\end{aligned}$$

Order Notation: $f(n) = O(g(n))$ if there are constants c, n_0 such that for all $n > n_0$, $f(n) \leq cg(n)$. $g(n)$ is said to be an **asymptotic upper bound** of $f(n)$. $f(n) = \Omega(g(n))$ if there are constants c, n_0 such that for all $n > n_0$, $f(n) \geq cg(n)$. $g(n)$ is said to be an **asymptotic lower bound** of $f(n)$.

$f(n) = o(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.] $\forall \epsilon\ \exists n_0\ \forall n \geq n_0$ $g(n) \geq \epsilon f(n)$.

Lemma 2. *The following containments hold.*

1. DTIME$(T(n)) \subseteq$ NTIME$(T(n))$ *and* DSPACE$(S(n)) \subseteq$ NSPACE$(S(n))$. Det TM are special NTM.
2. DTIME$(T(n)) \subseteq$ DSPACE$(T(n))$. You cannot use more memory than # steps.
3. *For* $S(n) \geq \log n$, DSPACE$(S(n)) \subseteq$ DTIME$(2^{O(S(n))})$. # config of TM $S(n)$-space bounded $\leq 2^{O(S(n))}$
4. NTIME$(T(n)) \subseteq$ DSPACE$(T(n))$. Outer det simulation of NTM
5. For $S(n) \geq \log n$, NSPACE$(S(n)) \subseteq$ DTIME$(2^{O(S(n))})$
6. For $S(n) \geq \log n$, NSPACE$(S(n)) \subseteq$ DSPACE$(S(n)^2)$



Lemma 3. *For* $S(n) \geq \log n$, NSPACE$(S(n)) \subseteq$ DTIME$(2^{O(S(n))})$.

$A \in NSPACE(S(n)) \Rightarrow \exists$ NTM N that is $S(n)$-space bounded and $A = L(N)$.

On input x, # configuration of $N \leq 2^{O(S(n))}$ where $n = |x|$

[We can show that $NSPACE(S(n)) \subseteq NTIME(2^{O(S(n))}) \subseteq DTIME(2^{2^{O(S(n))}})$

Pf: If N has an accepting computation on x, then it has an accepting comp. of length $\leq 2^{O(S(n))}$]

Configuration graph of N on x: $G_{N,x} = (V, E)$: V = Configurations of N on x

$(c_1, c_2) \in E$ iff $c_1 \vdash^1_N c_2$

$|V| \leq 2^{O(S(n))}$ $(n = |x|)$

Det Algo for : Given x, determine if N accepts x.

$\Longleftrightarrow$ determine if $\exists$ path from initial config of N on x to some accepting configuration in $G_{N,x}$.

Det Algo for A: Run DFS/BFS on $G_{N,x}$.

- Determine edges in $G_{N,x}$ by simulating N for one step.

Running Time $= S(n)\, 2^{O(S(n))}\, 2^{O(S(n))} = 2^{\log S(n) + 2O(S(n))} = 2^{O(S(n))}$

Theorem 4 (Savitch). *Let* $S(n) \geq \log n$. *Then* NSPACE$(S(n)) \subseteq$ DSPACE$(S(n)^2)$.

$A \in NSPACE(S(n)) \Rightarrow \exists$ NTM N that is $S(n)$-space bounded and $A = L(N)$.

Problem A : Given x, determine if N accepts x.

determine if $\exists$ path from initial config to accepting config in $G_{N,x}$.

vertices in $G_{N,x} \leq 2^{O(S(n))}$

If there is accepting path in $G_{N,x}$ then there is a path $\leq 2^{O(S(n))}$

Define (recursive) function PATH(c_1, c_2, k) s.t.

PATH(c_1, c_2, k) = true iff there is path from c_1 to c_2 in $G_{N,x}$ of length $\leq 2^k$.

```
def PATH (c1, c2, k)
  if k=0
    return (c1=c2) V c1 |-1_N c2
  for every c ∈ Vertex(G_N,x)
    if PATH(c1,c,k-1) ∧ PATH(c,c2,k-1)
      return true
  return false
```

for every accepting config c_A.

PATH$(c_0, c_A, O(S(n)))$

local vars = 4 and each is string $\leq O(S(n))$

stack depth $\leq O(S(n))$

Memory $\leq O(S(n))\, O(S(n)) = O(S(n)^2)$

$L \subseteq NL \subseteq P \subseteq NP \subseteq \begin{matrix} PSPACE \\ = \\ NPSPACE \end{matrix} \subseteq EXP \subseteq NEXP \subseteq \begin{matrix} EXPSPACE \\ = \\ NEXPSPACE \end{matrix}$

Common Complexity Classes:

$\mathrm{L} = \mathrm{DSPACE}(\log n)$	$\mathrm{NL} = \mathrm{NSPACE}(\log n)$
$\mathrm{P} = \mathrm{DTIME}(n^{O(1)})$	$\mathrm{NP} = \mathrm{NTIME}(n^{O(1)})$
$\mathrm{PSPACE} = \mathrm{DSPACE}(n^{O(1)})$	$\mathrm{NPSPACE} = \mathrm{NSPACE}(n^{O(1)})$
$\mathrm{EXP} = \mathrm{DTIME}(2^{n^{O(1)}})$	$\mathrm{NEXP} = \mathrm{NTIME}(2^{n^{O(1)}})$
$\mathrm{EXPSPACE} = \mathrm{DSPACE}(2^{n^{O(1)}})$	$\mathrm{NEXPSPACE} = \mathrm{NSPACE}(2^{n^{O(1)}})$

Time Constructible Functions: A function $T : \mathbb{N} \to \mathbb{N}$ is said to be time constructible if $T(n) \geq n$ and there is a Turing machine M such that M on input x, outputs a string of length $T(|x|)$ in time $T(|x|)$.

Space Constructible Functions: A function $S : \mathbb{N} \to \mathbb{N}$ is said to be space constructible if there is a Turing machine M such that M on input x, outputs a string of length $S(|x|)$ in space $S(|x|)$.

Proposition 5 (Universal Simulation). *For any time constructible function T, there is a Turing machine U such that U runs in time $O(T(n) \log T(n))$ and $\mathbf{L}(U) = \{\langle M, x\rangle \mid M \text{ accepts } x \text{ in time } T(|x|)\}$.*

For any space constructible function S, there is a Turing machine U such that U runs in space $O(S(n))$ and $\mathbf{L}(U) = \{\langle M, x\rangle \mid M \text{ accepts } x \text{ in space } S(|x|)\}$.

Theorem 6 (Space Hierarchy). *If S, S' are space constructible functions such that $S(n) = o(S'(n))$ then $\mathrm{DSPACE}(S(n)) \subsetneq \mathrm{DSPACE}(S'(n))$.*