Chapter 9 Intractability

- A problem is called <u>intractable</u> if it has a solution in principle, but the solution requires too much time or space to be practical.

- SAT is believed to be intractable. Yet, no one has proved it so far.

- We will present problems that are <u>provably</u> intractable.

HIERARCHY THEOREMS

Recall that for a function $s: \mathbb{N} \to \mathbb{R}$

$$SPACE(s) = \{A \subseteq \Sigma^* : \text{there is an } O(s) \text{ space bounded DTM } M \text{ deciding } A\}$$

By Theorem 3.13 and its proof, we may assume that M has only one work tape, and a ROM input tape.

- Our intuition: for a faster growing function s, the class $SPACE(s)$ gets larger.
- For instance, we expect that

$$SPACE(n) \subset SPACE(n^2) \subset SPACE(n^3) \subset \ldots$$

- This intuition turns out to be correct, provided that s is "nice" in the sense that it is easy to compute.

DEFINITION 9.1

Let $s : N \to N$ where $s(n) \geq \lg n$.
We say that s is <u>space constructible</u> if the function

$$1^n \longmapsto s(n) \text{ in binary}$$

is computable in space $O(s(n))$.

<u>Examples</u>

- $s(n) = n^2$
  - let n be the length of input x
  - multiply n by itself.
  - print the result in binary

- $s(n) = 2^n$

$$\underbrace{1 \, \text{----} \, 1}_{n} \longmapsto \underbrace{10 \ldots 0}$$

---

s space constructible means that an $O(s)$ space bounded machine can "mark off" the space it will need in advance.

Theorem 9.3 (Space hierarchy theorem)

Let $f: N \to N$ be space constructible. Then there is a language A in $SPACE(f)$ that is not computable in space $o(f(n))$.

Example
- For each $k > 1$, there exists a language in $SPACE(n^k) - SPACE(n^{k-1})$.
- There is $A \in SPACE(\log^2 n) - L$

PROOF OF 9.3.

We describe A by an algorithm D that runs in space $O(f)$. It implements the diagonal method of Ch. 4. First, we give a rough draft of the algorithm.

On input w, a description $\langle M \rangle$ of TM M,

- mark off space $f(|w|)$
- simulate M on input w, as long as the simulation does not use more space than was marked off.
- REJECT if that simulation accepts, ACCEPT otherwise.

If M works in space $o(f)$, we expect that the simulation of $M(w)$ can be completed.

PROBLEM 1 The fact that M uses space $\ll f(n)$ might kick in only at inputs much larger than $\langle M \rangle$

Solution: do the simulation for all strings of form $x = \langle M \rangle 10^k$.

- We can recover $\langle M \rangle$ from such x.
- if k is large, then $o(f)$ "kicks in".

PROBLEM 2 M may go into an infinite loop.

Solution: If M runs in space $o(f(n))$, it uses time at most $2^{o(f(n))}$. So we add a "clock" which allows the simulation to run at most $2^{f(n)}$ steps.

PROBLEM 3 M may have an arbitrary tape alph. Γ. Our alphabet has to be fixed.

Solution: Simulate one tape cell of M by a block of d tape cells, where $d \in \mathbb{N}$ depends on M. If M runs in space $o(f)$, then this linear increase in space doesn't matter.

We are now ready to give the algorithm describing A. It runs in space $O(f(n))$.

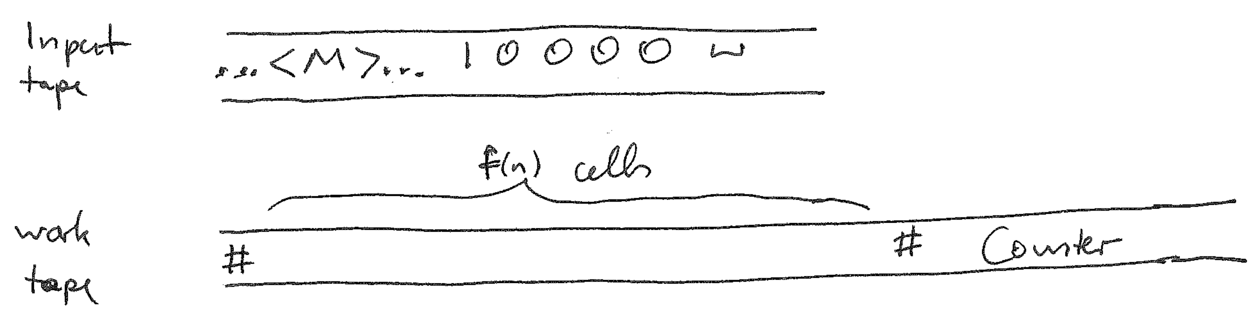
D = " on input w

1. let n be the length of w
2. Compute $f(n)$, and mark off this much tape. If later stages (of simulation) attempt to use more space, REJECT.
   Initialize counter to $1\overbrace{0\ldots0}^{f(n)}$
3. Check if w is of the form

$$\langle M\rangle 10^*$$

   for some TM M; else REJECT.
4. Simulate M on w, while counting down the number of steps used in the simulation. If counter reaches 0, REJECT
5. If M accepts, REJECT. If M rejects, ACCEPT."

Memory allocation of D



Verification

- D is a decider because each stage runs for a finite amount of steps. Let A be its language.
- D uses $O(f(n))$ work space, so $A \in SPACE(f)$

Suppose for a contradiction that TM M decides A in space $g(n)$, where $g(n) = o(f(n))$.

▷ There is a constant d such that D can simulate M (on all inputs) in space $d\,g(n)$.

Hence,

▷ there is n_0 such that

$$\forall n \geq n_0 \quad d\,g(n) < f(n).$$

▷ So, if $|w| \geq n_0$, the simulation in stage 4 is completed.

▷ Let $w = \langle M \rangle 1 0^{n_0}$.

Then

$$D \text{ accepts } w \leftrightarrow M \text{ rejects } w,$$

contrary to the assumption that M decides A. $\square$

<u>Corollary 1</u> Let $0 < p < q$, $p, q \in \mathbb{Q}$.

Then

$$SPACE(n^p) \subset SPACE(n^q).$$

<u>Proof</u>

- $n^p / n^q = n^{p-q} \xrightarrow[n \to \infty]{} 0$.

  Thus $n^p = o(n^q)$

- $n \to n^q$ is space constructible:

  write $q = r/s$, $r, s \in \mathbb{N}$.

  Then $n^q = \sqrt[s]{n^r}$

  $$= \max\{k : k^s \leq n^r\}$$

  This k can be computed in space $O(\lg n)$, which is $O(n^q)$

⑨

Corollary 2

$$NL \subset PSPACE$$

(so, at least one of the open problems

$$NL \overset{?}{\subset} P$$

$$P \overset{?}{\subset} PSPACE$$

has an affirmative answer!)

Proof
- By Savitch's Theorem,

$$NL = NSPACE(\lg n) \subseteq SPACE(\lg^2 n)$$

- The space hierarchy theorem shows that

$$SPACE(\lg^2 n) \subset SPACE(n),$$

because $n \to n$ is space constructible, and $\lg^2 n = o(n)$.

An explicit intractable problem A.

Let $f(n) = 2^n$.

- f is space constructible
- for each k, we have

$$n^k / 2^n \xrightarrow[n \to \infty]{} 0$$

i.e. $n^k = o(2^n)$.

Thus, $PSPACE \subset SPACE(2^n)$, where we defined the set A separating the two classes in the proof of Theorem 9.3.

Corollary 3 $\quad SPACE(n) \neq P$

Proof Let $A \in SPACE(n^2) - SPACE(n)$

Then $\tilde{A} = \{ x 1 0^{|x|^2} : x \in A \} \in SPACE(n)$.

Also, $A \leq_p \tilde{A}$ via $x \to x 1 0^{|x|^2}$.

If $SPACE(n) = P$, then we get $A \in P = SPACE(n)$, as P is closed downward under $\leq_p$. Contradiction.