Chapter 8.2 : The Class P-Space

SPACE COMPLEXITY is kind of like time complexity . . . but space is reused
describes the MAX # of tape cells looked at by a T.M.

Say there's a TM that checks whether an input is made of only odd #s.

T.M. M

Input (size = n) 3 1 5 9 3 7 2 1 9 . . .

Tape 1 1 1 1 1 1 0 1 1 . . .

The tape stores a boolean value, 1 if odd number, 0 if even number
Then M scans the tape, with n cells, to find a 0. If it does, REJECT, else ACCEPT the string.

So M will scan through at most n cells, and "runs in space $O(n)$"

Just like time-complexity classes, we have

$$SPACE(f(n)) = \{\text{languages decided by a DTM in } O(f(n)) \text{ space}\}$$

$$NSPACE(f(n)) = \{\text{languages decided by an NTM in } O(f(n)) \text{ space}\}$$

But unlike time-complexity classes, simulating an NTM using a DTM doesn't require an exponential increase !!
This is shown using Savitch's Theorem.

Abbreviation
- TM - Turing Machine
- DTM - Deterministic Turing Machine
- NTM - Non deterministic Turing Machine

Savitch's Theorem

An NTM using $\ell(n)$ space $\Rightarrow$ A DTM using $\ell^2(n)$ space [at most]

For any function $f: N \to N$ where $\ell(n) \geq \log n$: $NSPACE(\ell(n)) \subseteq SPACE(\ell^2(n))$

The idea : Recursively test the $\boxed{\text{yieldability problem}}$: get from the START to ACCEPT configuration of a TM in t steps by finding a middle configuration in $t/2$ steps.

$C_1 \xrightarrow{t} C_2$ $\Rightarrow$ uses $O(\ell(n))$ per level

$C_1 \xrightarrow{t/2} C_{m_1} \xrightarrow{t/2} C_2$

$C_1 \xrightarrow{t/4} C_{m_2} \xrightarrow{t/4} C_{m_1} \xrightarrow{t/4} C_{m_3} \xrightarrow{t/4} C_2$

⋮

$C_1 \xrightarrow{1} C_x \quad \ldots \quad C_y \xrightarrow{1} C_2$ $\Leftarrow$ Check each step when you reach 1 step between configurations

} $\log(t)$ levels $= O(\ell(n))$ from $t = 2^{O(\ell(n))}$

TOTAL SPACE
$O(\ell(n)) \times O(\ell(n))$
$= O(\ell^2(n))$

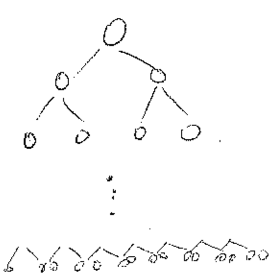
Now we have the space versions of PTIME and NPTIME :

$PSPACE = \bigcup_k Space(n^k)$ for languages decided in polynomial space by a DTM

$NPSPACE = \bigcup_k Space(n^k)$ for languages decided in polynomial space by an NTM

But by Savitch's Theorem, a language decided in $\ell(n)$ space by an NTM can also be decided by a DTM in $\ell^2(n)$ space.
A polynomial squared is still a polynomial : $\boxed{PSPACE = NPSPACE}$!!

Also: A TM using $\ell(n)$ space will have $\ell(n) \cdot 2^{O(\ell(n))}$ possible configurations.

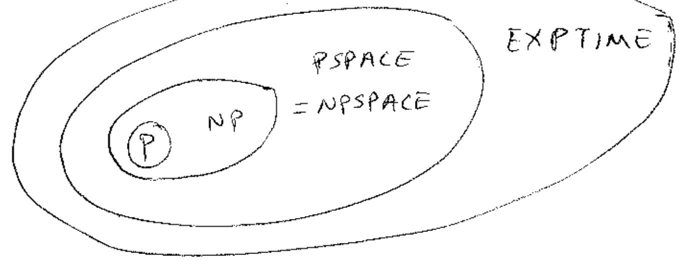


} $2^{O(\ell(n))}$ paths, each using maximum $\ell(n)$ space

Thus, it will run in

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

So we see the following :



MAIN POINT $\to$ Nondeterminism might save time, but not really space (for polynomial factors)