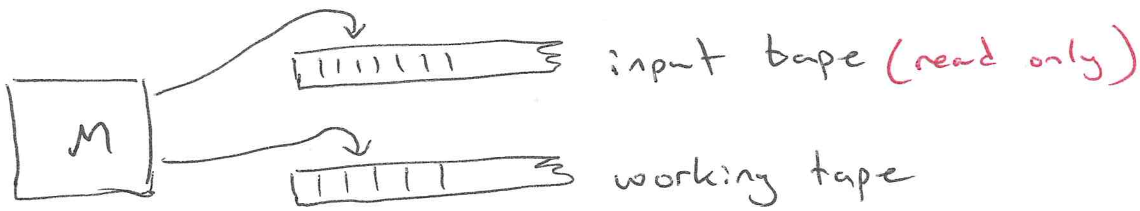
Space Complexity of TMs

Let M be a decider for some L

Assume M is a 2-tape TM



Def

The space complexity of M is a function $f: \mathbb{N} \to \mathbb{N}$ where $f(n)$ is the maximum number of work-tape cells visited over all inputs w with $|w| = n$.

Def

Let $f: \mathbb{N} \to \mathbb{R}^+$ be a function. The space complexity class is defined as

$$\text{SPACE}(f(n)) = \{ L \mid L \text{ is a language decided by a } O(f(n)) \text{ space deterministic TM} \}$$

If M is a non-deterministic TM the

$f(n)$ = the max number of work tape cells over all branches

$NSPACE(f(n)) = \{ L \mid L$ is a language decided by a $O(f(n))$ space non-deterministic TM $\}$

$$PSPACE = \bigcup_{k \geq 0} SPACE(n^k)$$

$$NPSPACE = \bigcup_{k \geq 0} NSPACE(n^k)$$

<u>Claim</u>

PSPACE $\subseteq$ EXP

where $EXP = \bigcup_{k > 0} TIME(2^{n^k})$

How can we even relate these complexity classes?

PSPACE allows use of at most $p(n)$ space but any amount of time

EXP allows use of $2^{p(n)}$ time and any amount of space

PSPACE $\subseteq$ EXP

How much time can we use for a PSPACE problem?

If we write to $p(n)$ tape cells there are $2^{p(n)}$ different string that can appear on the tape. Assuming $\Gamma = \{0,1\}$

The tape head can be in $p(n)$ different places.

The TM can be in one of k states.

The total number of configurations is

$$T(n) = k\, p(n)\, 2^{p(n)}$$

If we run M for $T(n)+1$ steps we must visit the same configuration twice. Since by definition M is deterministic that means it will loop and visit that configuration infinitely often.

Since PSPACE is defined for deciders any machine that loops doesn't solve a PSPACE problem.

PSPACE is the class of problems that use at most $p(n)$ space and at most $k\,p(n)\,2^{p(n)}$ time.

$$K \cdot p(n) \cdot 2^{p(n)} \leq 2^{q(n)}$$

for some polynomial q

therefore PSPACE $\subseteq$ EXP

How much space can we use for an EXP problem?

Since we are allowed $2^{p(n)}$ steps the tape head can't visit more than $2^{p(n)}$ tape cells.

PSPACE and EXP are both classes of problems solvable in exponential time.

- PSPACE is restricted to polynomial space use
- EXP is restricted to exponential space use.

$$\left.\begin{array}{l} P \subseteq PSPACE \\ NP \subseteq NPSPACE \\ PSPACE \subseteq NPSPACE \end{array}\right\} \text{trivial}$$

$PSPACE \subseteq EXP$

$$P \subseteq NP \subseteq \begin{array}{c} PSPACE \\ \shortparallel \\ NPSPACE \end{array} \subseteq EXP$$

Savitch's Theorem

Let N be a NTM deciding L using space $f(n)$ on inputs of length n. Then $\exists$ TM M that decides L using $O(f^2(n))$ space.

$$NPSPACE \subseteq PSPACE$$

A direct approach similar to what we did with time would produce an exponential increase in space requirements.

To get around this we will define a recursive method.

- Assume that N has a single accepting configuration C_{ACCEPT}

- Define recursive algorithm REACH(c_a, c_b, s)

configuration A   config B   # of steps

REACH(C_a, C_b, S) = 1 iff you can reach config C_b from C_a in at most S steps as determined by the transition function for N.

REACH(C_a, C_b, S)

if $s = 1$

return 1 if $C_a = C_b$ or C_b is reachable from C_a by one application of N's transition function.

if $s > 1$

For all possible configs C_z of machine N that use at most $f(n)$ space

① REACH(C_a, C_z, $S/2$)

② REACH(C_z, C_b, $S/2$)

if 1 and 2 return 1 RETURN 1

else RETURN 0

- M on input ω

Compute REACH(C_{start}, C_{accept}, $2^{kf(n)}$)

When REACH invokes itself recursively

it stores C_a, C_b, and s

this uses $O(f(n))$ space

initially $s = 2^{k f(n)}$ and each level divides

the steps in half.

so the depth of the recursion

is $O(\log(2^{k f(n)}))$ or $O(f(n))$

Therefore M uses $O(f^2(n))$ space

One problem is we need to know the value of

$f(n)$ before we begin

since we don't care about running time

we can just try all possible values one at

a time

PSPACE-Completeness

L is PSPACE-Complete if

① $L \in$ PSPACE

② $\forall A \in$ ~~N~~PSPACE, $A \leq_p L$

TQBF = $\{ \langle \phi \rangle \mid \phi$ is a true fully quantified Boolean formula $\}$

$\phi = \forall x \exists y ((x \vee y) \wedge (\bar{x} \vee \bar{y}))$

each variable must appears within the scope of some quantifier.

TQBF $\in$ PSPACE

M = "On input $\langle \phi \rangle$:

1. If ϕ contains no quantifiers, it must consist of only constants. Evaluate ϕ and ACCEPT if true Reject otherwise.
2. If $\phi = \exists x_1 \psi$
   call M on $\langle \psi \rangle$ with $x_1 = 0$ everywhere
   call M on $\langle \psi \rangle$ with $x_1 = 1$ everywhere
   ACCEPT if either call accepts
3. If $\phi = \forall x_1 \psi$
   call M on $\langle \psi \rangle$ with $x_1 = 0$ everywhere
   call M on $\langle \psi \rangle$ with $x_1 = 1$ everywhere
   ACCEPT if both accept.

$$\text{EXPTIME} = \bigcup_{c>0} \text{TIME}(2^{n^c})$$

$$P \subseteq NP \subseteq PSPACE \subseteq EXPTIME$$

$$P \subset EXPTIME$$

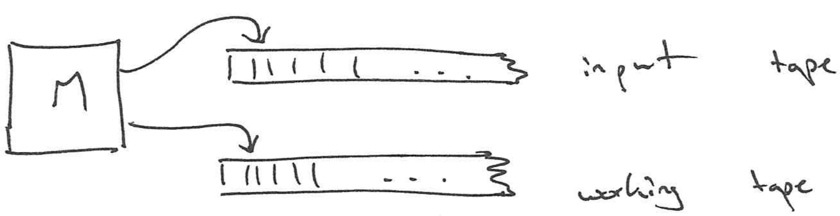
but that's all we can prove

Space - complexity of TMs

Def

Let M be a decider for some L

Assume M is a 2-tape TM



The space complexity of M is a function $f: \mathbb{N} \to \mathbb{N}$ where $f(n)$ is the maximum number of work-tape cells visited, over all inputs w with $|w| = n$.

Space Complexity Class

$$\text{SPACE}(f(n)) = \{ L \mid L \text{ is decidable using space } O(f(n)) \}$$

$$\text{PSPACE} = \bigcup_{k \geq 0} \text{SPACE}(n^k)$$

If M is a NTM then $f(n)$ = max number of work cells over all branches.

$$\text{NPSPACE} = \bigcup_{k \geq 0} \text{NSPACE}(n^k)$$

$$\text{PSPACE} = \text{NPSPACE}$$

$$\text{PSPACE} \subseteq \text{EXPTIME}$$