LOGSPACE Complexity

$$L = SPACE(\log n)$$

The class of all problems decidable using $O(\log n)$ additional space.

- Why $\log n$?
- Why log space and not log time?

$L \subseteq P$

$$SPACE(f(n)) \subseteq TIME(2^{f(n)})$$

This was used to show PSPACE $\subseteq$ EXP

is $L = P$? unknown

generally believed $L \subset P$

$EVEN = \{w \mid w \text{ contains an even number of 1s}\}$

Is $EVEN \in L$? yes

we need a finite amount of space

i.e. constant.

$A = \{0^k 1^k \mid k \geq 0\}$

is $A \in L$? yes

we can count the number of 0's using

$\log k$ space.

Nondeterministic LOGSPACE

$NL = NSPACE(\log n)$ Def #1

What if I want a certificate based definition?

The idea for the NP definition was to use the path through the NTM to create the certificate.

Here we might generate a certificate that is too long to store.

To get around this we allow for a read-once certificate tape.

A language A is in NL if there exists

- a deterministic TM M using at most $O(\log n)$ space for every input w where $|w| = n$
- M has a certificate tape

$\forall\ w \in \{0,1\}^*$

$w \in A \iff \exists\ c \in \{0,1\}^{n^k}$ such that $M(w,c) = 1$

$PATH = \{\langle G, s, t\rangle \mid G$ is a directed graph with a path from s to $t\}$

Is this in P? yes

is PATH $\in$ NL? yes

is PATH $\in$ L? not that we know of

PATH $\in$ NL

- start at node s
- nondeterministically the next node in the path
- if t is reached ACCEPT
- if $|V|$ nodes are visited REJECT

This requires storing the current node and the number of nodes visited

- current node $O(1)$
- path length $O(\log n)$

$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq EXP$

LogSpace Reductions

Suppose we have two languages A and B.

$$A \leq_L B$$

Since all problems in NL are solvable in poly-time we can't use a poly-time reduction.

Log space transducer

A TM with a read-only input tape, a read/write work tape, and a write-only output tape. The work tape may contain $O(\log n)$ symbols.

A log space transducer M computes a function $f: \Sigma^* \to \Sigma^*$ where $f(w)$ is the string remaining on the output tape when M halts on input w.

If $A \leq_L B$ and $B \in L$ then $A \in L$

can we use the same approach we used for poly-time reductions?

No $f(w)$ may be too large to fit in log-space.

Instead we will compute symbols in $f(w)$ only when requested.

- let M_A be a log space decider for A and M_B for B

Every time M_B needs to read a symbol for $f(w)$. The computation of $f(w)$ will be restarted until the needed symbol is output.

Trading time for space

NL-Completeness

A language B is NL-Complete if

1. $B \in NL$
2. $\forall A \in NL \quad A \leq_L B$

PATH is NL-Hard

For any language $A \in NL$ construct a graph that represents the configurations of A's NTM on some input w.

- let NTM M decide A in $O(\log n)$ space
- Given input w construct $\langle G, s, t \rangle$ in log space where G contains a path from s to t iff M accepts w
- Each node ~~tag~~ in G is a configuration of M on w.
- For each pair of configurations C_1 and C_2 $(C_1, C_2) \in E$ if C_2 is one of the possible next configurations of M starting at C_1

- Node s is the start configuration of M on w
- Node t is the accepting configuration of M on w.

PATH is NL-Complete

NL = Co-NL

Time Hierarchy Theorem

For any time constructible function $t: \mathbb{N} \to \mathbb{N}$ a language A exists that is decidable in $O(t(n))$ time but not decidable in time

$$o\left(\frac{t(n)}{\log(t(n))}\right)$$

Idea

Construct TM D that decides A in $O(t(n))$ time, whereby A cannot be decided in $o\left(\frac{t(n)}{\log t(n)}\right)$ time. D takes an input w of the form $\langle M \rangle 10^*$ and simulates M on input w, making sure not to use more than $t(n)$ time. If M halts within that time D outputs the opposite result.

We introduce a logarithmic factor overhead in running time.

Proof

D = "On input w:

1. let n be the length of w
2. Compute $t(n)$ ~~using~~ and store the value of $\left\lceil \frac{t(n)}{\log t(n)} \right\rceil$ in a binary counter.

   Decrement this counter before each step used in stages 4 or 5. If the counter ever hits 0 REJECT
3. If w is not of the form $\langle M \rangle 10^*$ for some TM M reject
4. Simulate M on w
5. If M accepts then REJECT
   If M rejects then ACCEPT

Assume that TM M decides A in time $g(n)$ where $g(n)$ is $o\left(\frac{t(n)}{\log t(n)}\right)$

D can simulate M using time $d\,g(n)$ for some constant d.

If the total time to simulate M is at most $\frac{t(n)}{\log t(n)}$ then the simulation will run to completion.

Because $g(n)$ is $o\left(\frac{t(n)}{\log t(n)}\right)$ some constant n_0 exists where $d\,g(n) < \frac{t(n)}{\log t(n)}$ for all $n \geq n_0$

Therefore D's simulation of M will run to completion as long as the length of w is n_0 or more

Run D on $\langle M \rangle 10^{n_0}$