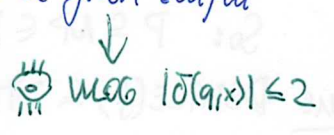


Non-deterministic TM (NTM)

extend $\delta: Q \times \Gamma^k \rightarrow \mathcal{P}(Q \times \Gamma^k \times \{\leftarrow, \rightarrow, \circ\}^k)$... choice of instruction from the set

- successor relation is not a function \rightarrow multiple computations for a given output
- if $\delta(q,x) = \emptyset$, we assume rejection.
- input α accepted $\equiv \exists$ at least one accepting computation
- halting \equiv all computations halt
- time & space: maximum over all computations



👁 enumeration works again ($NM_\alpha = NTM$ with code α), we have an universal NTM d.c.

Exercise: k-tape \rightarrow 2-tape NTM with only constant-factor slowdown

Df: Non-deterministic complexity classes: $NTIME(f)$, $NTIMEF(f)$ for functions

Theorems $NP = NTIME(poly(n)) = \bigcup_{k \geq 0} NTIME(n^k)$

↳ all accepting computations must agree on result, \exists at least one accepting comp.

Proof: \subseteq the NTM guesses the certificate using non-determinism & then it runs the verifier

\supseteq the certificate encodes the non-deterministic choices

Example: The following problem is NP-complete:

$\{ \langle \alpha, \beta, t \rangle \mid NM_\alpha \text{ accepts input } \beta \text{ within } t \text{ steps} \}$

NP: simulate $NM_\alpha(\beta)$ using universal NTM with an "alarm clock" (reject after t steps)

reduction: calculate $t \in poly(n)$, $\alpha :=$ code of NTM solving source problem pass α, β

Space Complexity

We want to count only "work space" of the TM.

3 types of tapes:

- input tape: read-only, head doesn't move more than 1 cell before/after input string
- k work tapes: read-write
- output tape: write-only, head cannot move left

space used by computation \equiv # visited cells on the work tapes (for NTM: max over computations)

👁 This doesn't change time complexity classes: we can copy input \rightarrow work \rightarrow output with constant slowdown

👁 We can encode information in position of head on input tape.

- this makes a difference if work space $\in O(\log n)$
- otherwise we can keep track of the head position in binary

Space classes

- $DSPACE(f)$ } decision problems
- $NSPACE(f)$ }
- & $DSPACEF(f)$ } functions
- $NSPACEF(f)$ }
- $PSPACE = DSPACE(poly(n))$
- $NPSPACE = NSPACE(poly(n))$

We want f to be:

- 1 non-decreasing
- 2 space-constructible
↳ $f(n)$ can be computed from 1^n , result in binary, in space $O(f(n))$
- 3 usually $f(n) \geq \log n$

proper space-complexity function

Basic Inclusions: $DTIME(f) \subseteq NTIME(f) \subseteq DSPACE(f) \subseteq NSPACE(f)$
 \uparrow Can try all certificates in space $O(f)$

So: $P \subseteq NP \subseteq PSPACE \subseteq NPSPACE$

Thm: $DSPACE(f) \subseteq DTIME(2^{O(f)})$ for every $f \geq \log n$.

Proof: First, let's bound # reachable configurations: $|Q| \cdot (n+2) \cdot (|\Sigma|+1)^{f(n)} \cdot f(n)^k$
 \uparrow state \uparrow pos. of head on input tape \uparrow contents of work tapes, extra character for "end of tape" \uparrow pos. of heads on work tapes \leftarrow # tapes

... this is $O(2^{O(f(n))})$

If a configuration repeats, the whole computation loops. (This requires deterministic TM)
 \Rightarrow add a binary counter of $O(f(n))$ bits, use it as alarm clock (increment in every step of the original TM). Alarm expires \Rightarrow reject.

In space-bounded computation with space $\geq \log n$, we can always make sure that the machine halts.

Corollary: $NPSPACE \subseteq EXPTIME := DTIME(2^{O(n)})$.

We want to prove the same for $NSPACE(f)$, but \otimes makes it more complicated.

Reachability method

\swarrow within space $f(n)$

Def: Configuration graph of a given NTM on a given input is a directed graph with:
 $V :=$ set of configurations (for input tape, consider only head position)
 \uparrow limited by available space
 $E :=$ successor relation

start $\in V$... initial config

accept $\in V$... modify the TM to clean up before accepting $\left\{ \begin{array}{l} \text{clear working tapes} \\ \text{rewind input tape} \end{array} \right\}$ unique accepting config

$|V| \in O(2^{O(f)})$, $|E| \in O(|V|)$, graph can be generated in $O(\text{poly}(|V|))$ time & $O(f)$ space.

Machine accepts \Leftrightarrow graph contains a (directed) path from start to accept.

Thm: $NSPACE(f) \subseteq DTIME(2^{O(f)})$ for every $f \geq \log n$. [therefore $NPSPACE \subseteq EXPTIME$]

Proof: Construct the reachability graph & run BFS on it.
 \uparrow time $O(\text{poly}(|V|, |E|))$ \uparrow also time $O(\text{poly}(|V|, |E|))$
 \uparrow which is $O(2^{O(f)})$

Generally: Time-/space-efficient algorithms for REACH translate to inclusions of complexity classes.
 $\uparrow \{G, s, t\} \mid \exists \text{ path from } s \text{ to } t \text{ in } G$

Thm (Savitch's): $NSPACE(f) \subseteq DSPACE(f^2)$ for every $f \geq \log n$.

Corollary: $NPSPACE \subseteq PSPACE$, so $NPSPACE = PSPACE$.
 \rightarrow will be proven soon...

Lemma: REACH $\in O(\log^2 n)$

Proof: Use "middle-first search".

Recursive function $D_k(x,y)$ computing " \exists walk from x to y with at most 2^k edges".

We have: $D_0(x,y) = (x=y) \vee ((x,y) \in E)$

$D_k(x,y) = \bigvee_{z \in V} (D_{k-1}(x,z) \wedge D_{k-1}(z,y))$... FOR loop & recursion

Every level of recursion requires $O(\log n)$ space for local variables, $\log n$ levels suffice to find a path.

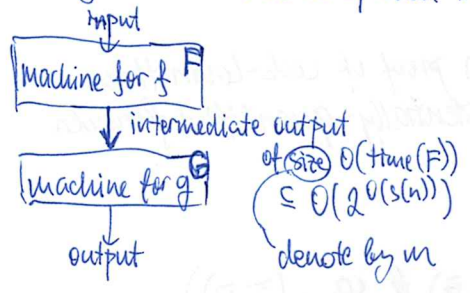
Now, we want to combine generator of config graph with this algorithm, but we don't have space to store the graph.

Lemma: If f can be computed in space $s(n)$ and g $\xrightarrow{||}$ $t(m)$,

Then $g(f(\dots))$ can be computed in space $O(s(n) + t(m))$

also applies to composition of a language with a function (that is, a reduction) $O(\log m)$ space

Proof:



Start G & keep track of (position) of head on G 's input tape. Whenever G moves its input head (& at the start of computation), re-run F to get the corresponding symbol of its output.

\hookrightarrow modify F : reset work tapes on startup, reset input head $\xrightarrow{||}$ keep track of output head position

Write to output tape: compare with G 's input head pos.
 = remember char in state
 \neq discard written char (won't be read by F)

Total space:

- $s(n)$ for F
- $O(\log m)$ for head positions ... this is $O(t(m))$
- $t(m)$ for G

Corollary: Savitch's thm.

\hookrightarrow If $L \in \text{NSPACE}(f)$: graph generation requires $O(f)$ space, reachability needs $O((2^{O(f)})^2) = O(2^{O(f)})$

combined by the lemma to $O(2^{O(f)})$

Remark: REACH $\in O(\log n)$ would imply $\text{NSPACE}(f) = \text{DSPACE}(f)$... but this is long open.

It's known that undirected $\text{UREACH} \in O(\log n)$ [Reingold 2004, non-trivial]

\hookrightarrow this implies only $\text{SSPACE}(f) = \text{DSPACE}(f)$

\uparrow symmetric non-determinism (successor relation symmetric)

So we have: $\text{DTIME}(f) \subseteq \text{NTIME}(f) \subseteq \text{DSPACE}(f) \subseteq \text{NSPACE}(f) \subseteq \text{DTIME}(2^{O(f)}) \subseteq \text{DSPACE}(f^2)$

and: $\text{NSPACE}(\log n) \subseteq P \subseteq NP \subseteq \text{co-NPSPACE} = \text{NPSPACE} \subseteq \text{EXPTIME}$

\uparrow this is also known as NL

\uparrow also = co-NPSPACE as PSPACE is closed under complements