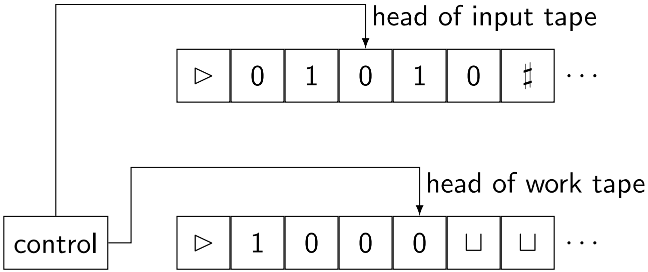
Chapter 4 - Inside **P**

- Turing machines with input and output tape
- space complexity below linear
- complexity classes **L**, **NL**, and co-**NL**
- paths in directed graphs and **NL**
- log space reductions and **NL**-complete problems
- **NL** = co-**NL** (Theorem of Immerman/Szelepcsènyi)

Turing machines with input tape

- A Turing machine with input tape is a 2-tape TM. The first tape is called the *input tape*, the second tape is called the *work tape*.
- The first tape is a read-only tape. At the beginning of a computation the input tape contains the input w. The end of the input is marked by the special symbol $\sharp$.
- The TM cannot change any symbols on its input tape. Furthermore, its head cannot move beyond the symbol $\sharp$.
- The work tape can be read and written in the usual way.
- Configurations of TM M on input w consist of
  1. the position *pos* of the head on the input tape,
  2. the content of the work tape (ignoring $\sqcup$'s as usual) and the position of the head of the work tape,
  3. the state.

Schematic of a Turing machine with input tape



Space complexity of TMs with input tape

Definition 4.1

Let M be a TM with input tape that halts on all inputs. The space complexity of M is the function $f : \mathbb{N} \to \mathbb{N}$, where $f(n)$ is the maximum number of tape cells on its work tape that M scans on any input of length n.

Definition 4.2

Let $s : \mathbb{N} \to \mathbb{R}^+$ be a monotonically increasing function. The space complexity class $\mathbf{DSPACE}\big(s(n)\big)$ consists of all languages that are decidable by an $\mathcal{O}\big(s(n)\big)$ space DTM with input tape. Similarly, the class $\mathbf{NSPACE}\big(s(n)\big)$ consists of all languages that are decidable by an $\mathcal{O}\big(s(n)\big)$ space NTM with input tape.

Classes **L** and **NL**

Definition 4.3

L *is the class of languages that are decidable in logarithmic space on a DTM with input tape, i.e.*

$$\mathbf{L} := \mathbf{DSPACE}\big(\log(n)\big).$$

Definition 4.4

NL *is the class of languages that are decidable in logarithmic space on a NTM with input tape, i.e.*

$$\mathbf{NL} := \mathbf{NSPACE}\big(\log(n)\big).$$

A language in **L**

Example

The language $\{0^k 1^k | k \in \mathbb{N}\}$ is a member of **L**.

$M =$ "On input $w \in \{0,1\}^*$:

1. Scan w to test whether w is of the form $0^n 1^m$. If not, *reject*.
2. Count the number of 0's and 1's in w. If these numbers are the same *accept*, otherwise *reject*."

A language in **NL**

Example

The language

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

is a member of **NL**.

An NTM deciding PATH

$N =$ "On input $G = (V, E)$ directed graph and $s, t \in V$:

1. Compute $|V| - 1$. Set $i \leftarrow 0, v \leftarrow s$.
2. Repeat until $i = |V| - 1$ or N has accepted:
3. Nondeterministically select $(v, u) \in E$.
4. If $u = t$, then *accept*.
5. Set $i \leftarrow i + 1, v \leftarrow u$.
6. *Reject*."

Classes **P** and **NL**

Theorem 4.5

$\mathbf{NL} \subseteq \mathbf{P}$.

Configurations

Configurations of TMs with input tape

For a TM with input tape and $w \in \Sigma^*$, a *configuration of M on w* is a setting of the state, the work tape, and the positions of the two tape heads. The input w is not part of a configuration of M on w.

Observation

If M is a TM with input tape that runs in space $s(n)$, then there is a constant c such that for an input w of length n the number of configurations of M on w is bounded by $n \cdot 2^{c \cdot s(n)}$.

Proof of Theorem 4.5

- Let L be a language in **NL** and M be a $\mathcal{O}(\log(n))$ space NTM with $L = L(M)$. Consider $w \in \Sigma^*, |w| = n$ arbitrary.
- The number of configurations of M on w is $(n+2) \cdot 2^{c \cdot \log(n)} \leq n^{c+2}$, which is polynomial.
- Let G be the configuration graph of M on input w and let s be the starting configuration of M on input w.
- G has polynomial size.
- We may assume that M has a single accepting configuration t.
- $w \in L(M)$ if and only if there is a directed path from s to t in G.
- We can decide in polynomial time whether $w \in L$ by using breadth-first-search on graph G and vertices s and t.

$\Rightarrow$ $L \in$ **P**.

Deterministic and nondeterministic space

Theorem 4.6 (Savitch's theorem - general version)

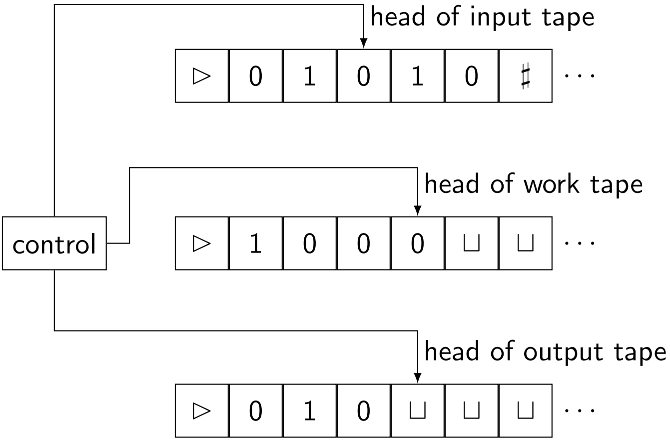
Let $s : \mathbb{N} \to \mathbb{N}$ *be a space constructible function with* $s(n) \geq \log(n)$ *for all* $n \in \mathbb{N}$, *then*

$$\mathbf{NSPACE}(s(n)) \subseteq \mathbf{DSPACE}(s(n)^2).$$

Turing machines with input and output tape

- A Turing machine M with input and output tape is a 3-tape TM. The first tape is called the *input tape*, the second tape is called the *work tape*, the third tape is called *output tape*.
- The input tape and the work tape are as for Turing machine with input tape only.
- The output tape is a write-only tape, i.e.
  1. the head of the tape can only move to the right
  2. if the head writes a symbol, it moves one cell to the right.
- Configurations of TM M on input w consist of
  1. the position *pos* of the head on the input tape,
  2. the content of the work tape (ignoring $\sqcup$'s as usual) and the position of the head of the work tape,
  3. the state.
- Neither the content nor the head position of output contribute to a configuration.

Schematic of a Turing machine with input & output tape



Space complexity of TMs with input & output tape

Definition 4.7

Let M be a TM with input and output tape that halts on all inputs. The space complexity of M is the function $f : \mathbb{N} \to \mathbb{N}$, where $f(n)$ is the maximum number of tape cells on its work tape that M scans on any input of length n.

Definition 4.8

A function $f : \Sigma^* \to \Sigma^*$ is a log space computable function if a deterministic $\mathcal{O}(\log(n))$ space Turing machine M with input and output tape exists that halts with $\triangleright f(w)$ on its output tape, when started on any input $w \in \Sigma^*$.

Log space reductions

Definition 4.9

Language A is log space reducible to language B, written $A \leq_L B$, if a log space computable function $f : \Sigma^* \to \Sigma^*$ exists, where for every $w \in \Sigma^*$

$$w \in A \Leftrightarrow f(w) \in B.$$

Lemma 4.10

Let f and g be log space computable functions. Then $g \circ f$ is log space computable.

Corollary 4.11

If $A \leq_L B$ and $B \leq_L C$, then $A \leq_L C$.

Proof of Lemma 4.10

- Let M_f, M_g be log space DTMs with input and output tape that compute f, g, respectively.

The problem

On input w cannot compute $f(w)$ with M_f and then $g(f(w))$ using M_g, since we may not be able to store $f(w)$ in space $\mathcal{O}(\log(n))$ on work tape.

The solution

$M_{g \circ f} =$ "On input w:

1. Run M_g on input $f(w)$. Whenever M_g needs to read a symbol of $f(w)$, run M_f with input w to compute that symbol, while ignoring all other symbols of $f(w)$."

NL-complete languages

Definition 4.12

A language B is **NL***-complete, if*

1. $B \in \mathbf{NL}$, *and*
2. *every language* $A \in \mathbf{NL}$ *is log space reducible to* B.

Theorem 4.13

If $A \leq_L B$ *and* $B \in \mathbf{L}$, *then* $A \in \mathbf{L}$.

Corollary 4.14

If any **NL***-complete language is in* **L**, *then* $\mathbf{L} = \mathbf{NL}$.

Paths in directed graphs and class **NL**

Theorem 4.15

PATH is **NL***-complete.*

Proof

- $A \in \mathbf{NL}$ and N a log space NTM with input tape and $L(N) = A$
- assume N has unique accepting configuration w_{accept}
- reduction f from A to PATH maps w to $\langle G, s, t \rangle$, where
    1. G is the configuration graph of N on input w
    2. s is the start configuration of N on input w
    3. $t = w_{\text{accept}}$

NL and co-**NL**

Theorem 4.16 (Immerman/Szelepcsènyi)

$\mathbf{NL} = \textit{co-}\mathbf{NL}$.

Very rough idea

- Construct log space NTM that decides the complement $\overline{\text{PATH}}$ of language PATH, where

$$\overline{\text{PATH}} := \{\langle G, s, t\rangle \mid G \text{ is a directed graph that does } \textit{not} \text{ have a directed path from } s \text{ to } t\}$$

Proof outline

- Construct log space NTM that decides the complement $\overline{\text{PATH}}$ of language PATH, where

$$\overline{\text{PATH}} := \{\langle G, s, t\rangle \mid G = (V, E) \text{ is a directed graph that does } \textit{not} \text{ have a directed path from } s \text{ to } t\}$$

- For $i = 0, \dots, |V| - 1$ set

$$A_i := \{v \in V \mid G \text{ has a directed path of length at most } i \text{ from } s \text{ to } v\}$$

- $c_i := |A_i|, i = 0, \dots, |V| - 1$

$\Rightarrow$ $(G, s, t) \in \overline{\text{PATH}} \Leftrightarrow t \notin A_{|V|-1}$

Proof outline

NTM for $\overline{\text{PATH}}$

- Inductively and nondeterministically compute the values c_i, $i = 0, \ldots, m$, where $m = |V| - 1$, i.e. for the computation of c_i only knowledge of c_{i-1} is required and if the NTM does not reject it has computed c_i correctly.
- Given c_m compute A_m nondeterministically, i.e. guess the elements v of A_m and verify nondeterministically that $v \in A_m$.
- Accept if and only if $t \notin A_m$.

An NTM deciding $\overline{\text{PATH}}$

$M =$ "On input $\langle G, s, t\rangle$, $G = (V, E), m = |V| - 1$:

1. Let $c_0 = 1$.
2. For $i = 0$ to $m - 1$:
3. Let $c_{i+1} = 1$.
4. For each node $v \neq s$ in G:
5. Let $d = 0$.
6. For each node u in G:
7. Nondeterministically either perform or skip these steps:
8. Nondeterministically follow a path of length at most i from s and *reject* if it does not end at u.
9. Increment d.
10. If $(u, v) \in E$, increment c_{i+1} and go to 5. with next v.
11. If $d \neq c_i$, then *reject*.

An NTM deciding $\overline{\text{PATH}}$

$M =$ "On input $\langle G, s, t \rangle$, $G = (V, E), m = |V| - 1$:

$\vdots$

12. Let $d = 0$.
13. For each node $u \neq t$ in G:
14. Nondeterministically either perform or skip
15. Nondeterministically follow a path of length at most m from s and *reject* if it does not end at u.
16. Increment d.
17. If $d \neq c_m$, then *reject*, otherwise *accept*. "