Complexity Theory

Two goals:

↳ Study computational models and programming constructs
in order to understand
their power and limitations

↳ Study computational problems wrt.
their inherent complexity

Usually, complexity means time and space requirements (on a particular model).
Can also mean other measures like
randomness, number of alternations, circuit size.

Background:

↳ Theory of computability goes back to
Piesburger, Church, Kleene, Post, Gödel, Turing, 1st half 20th century.

↳ Complexity Theory goes back to
Hartmanis, Stearns:
"On the computational complexity of algorithms"
(use multitype TM, argue that concepts apply
to any reasonable model of computation).

1. Lower Bounds via Crossing Sequences

Goal: Prove an unconditional lower bound on the power of a (uniform) complexity class.

Unconditional:
- Showing an NP-hardness result means "the problem is hard, assuming $P \neq NP$".
- Our result does not need such a condition.

Uniform:
- One algorithm (TM) for all inputs.
- Non-uniform models may use different algorithms (circuits) for different instances.

Note: Do not know other unconditional lower bounds, even proving $SAT \notin DTIME(n^3)$ seems out of reach.

Approach: Employ a counting technique called crossing sequences for proving lower bounds (vaguely related to pooling sets in automata theory).

1.1 Crossing Sequences

Let $COPY := \{w\#w \mid w \in \{a,b\}^*\}$.

The language is not context free (and hence not regular)

Goal:
- ↳ **Upper bound:** COPY can be decided in quadratic time
- ↳ **Lower bound:** COPY cannot be decided in subquadratic time (on a deterministic 1-tape TM).

Recall:
(1) When we refer to a problem written as a set,

deciding the problem means deciding membership for that set.

Given $x \in \{a,b\}^*$, does $x \in COPY$ hold?

(b) The time/space is measured relative to the size of the input. Without further mentioning, the size will be referred to as n.

(c) O-notation = asymptotic upper bound ($\leq$)
"no more than"

$$O(g(n)) := \{ f: \mathbb{N} \to \mathbb{N} \mid \exists c \in \mathbb{N}\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0 : f(n) \leq c \cdot g(n) \}$$

o-notation = asymptotic strict upper bound ($<$)
"less than"

$$o(g(n)) := \{ f: \mathbb{N} \to \mathbb{N} \mid \forall c \in \mathbb{R}^+\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0 : f(n) \leq c \cdot g(n) \}.$$

Claim: $COPY \in DTIME(n^2)$

We assume that all tapes are right-infinite.
To the left, they are marked by $\$$ (from there only move right, do not change)

Let M be a $t(n)$-time-bounded 1-tape TM for $COPY$ ($t: \mathbb{N} \to \mathbb{N}$).
We assume that M halts on the $\$$ marker (left).

Definition:
A crossing sequence of M on input x at position i is the sequence of states of M when moving its head from cell i to cell $i+1$ or from cell $i+1$ to cell i.
Denote the sequence by $CS(x,i)$.

Note: If q is a state in an odd position of the crossing sequence, then M moves its head from left to right.
If it is in an even position, M moves from right to left.

Lemma:

Let $x = x_1 . x_2$ and $y = y_1 . y_2$.

If $CS(x, |x_1|) = CS(y, |y_1|)$,

then $x_1 . x_2 \in L(M)$ iff $x_1 . y_2 \in L(M)$.

Proof:

Intuitively, the crossing sequence is all that M remembers about x_2 resp. y_2 when operating on the left part x_1.

Since the crossing sequences are assumed to coincide, M will behave the same on x_1, independent of whether x_2 or y_2 is on the right!

Since the # symbol is on the left, we are done. □

Theorem: $COPY \notin DTIME(o(n^2))$.

Proof:

Let M be a deterministic 1-tape TM for COPY.

Consider inputs of the form

$x = w_1 . w_2 \# w_1 . w_2$ with $|w_1| = |w_2| = n$.

For all $v \neq w_2$ with $|v| = |w_2|$, we have

$CS(x, i) \neq CS(w_1 . v \# w_1 . v, i)$ for all $2n+1 \leq i \leq 3n$.

Otherwise, M would accept

$w_1 . w_2 \# w_1 . v$ for some $v \neq w_2$ with $|v| = |w_2|$.

-3- by previous lemma.

We have

$$\boxed{Time_M(x) \geq \sum_{i \geq 0} |CS(x,i)|}$$ (equality holds if head is always moved)

With this,

$$\sum_{w_2 \in \{a,b\}^n} Time_M(w_1 w_2 \# w_1 w_2)$$

$$\geq \sum_{w_2} \sum_{v=2n+1}^{3n} |CS(w_1 w_2 \# w_1 w_2, v)|$$

$$= \sum_{v=2n+1}^{3n} \sum_{w_2} |CS(w_1 w_2 \# w_1 w_2, v)|.$$

- Fix some v with $2n+1 \leq v \leq 3n$.

For all w_2, the crossing sequences

$$CS(w_1 w_2 \# w_1 w_2, v)$$

are pairwise distinct (see above).

Let l_v be the average length of such a sequence, which means

$$\sum_{w_2} |CS(w_1 w_2 \# w_1 w_2, v)| = 2^n \cdot l_v.$$

- If l_v is the average length of a crossing sequence, at least half of the crossing sequences have length $\leq 2 l_v$ (see below).

There are

$$\leq (|Q|+1)^{2 \cdot l_v}$$

crossing sequences of length $\leq 2 l_v$ ($+1$ if stack is absent).

Hence,

$$\boxed{(|Q|+1)^{2 l_v} \geq \frac{2^n}{2}}$$

// at least half of the (different) crossing sequences
// have length $\leq 2 l_v$.

Thus, $l_v \geq c \cdot n$, for an appropriate $c > 0$, dependent on $|Q|$ (not on n and v)

(to enforce the inequality, we need that the crossing sequences are different)

- This yields

$$\sum_{w_2} \text{Time}_M(w_1 w_2 \# w_1 w_2) \geq \sum_{\nu=2n+1}^{3n} \sum_{w_2} |CS(w_1 w_2 \# w_1 w_2, \nu)|$$

$$= \sum_{\nu=2n+1}^{3n} 2^n \ell_\nu$$

$$\geq \sum_{\nu=2n+1}^{3n} 2^n \cdot c \cdot n$$

$$= 2^n \cdot c \cdot n^2.$$

- Since there are only 2^n words w_2, we get

$$\text{Time}_M(w_1 w_2 \# w_1 w_2) \geq c \cdot n^2$$

for at least one w_2. □

Lemma:

If $\frac{\sum_{i=1}^{n} w_i}{n} = d$, then at least half of the w_i have a value $\leq 2d$.

Proof:

Assume at least half of the w_i have a value $> 2d$.

Then

$$\sum_{i=1}^{n} w_i > \frac{n}{2} \cdot 2d = nd.$$

From this

$$\frac{\sum_{i=1}^{n} w_i}{n} > \frac{nd}{n} = d \quad \lightning \text{ to the assumption } \frac{\sum_{i=1}^{n} w_i}{n} = d$$

□

1.2 A Gap Theorem for Deterministic Space Complexity

Goal: Show that $o(\log\log n)$ work tape is no better than no tape at all.

Theorem: $DSPACE(o(\log\log n)) = DSPACE(O(1))$.

Inclusion $\supseteq$ can be solved as homework.

We show $\subseteq$. Note: The input tape is read-only.

Definition:
A small configuration of a TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$ consists of
- ↳ the current state (in Q)
- ↳ the content of the work tape (in Γ^*)
- ↳ and the head position of the work tape.

We neglect
- ↳ the input and
- ↳ the head on the input tape.

Definition:
The extended crossing sequence of M on input x at position i is the sequence of small configurations of M when moving its head from cell i to $i+1$ or from cell $i+1$ to i on the input tape.

Denote this sequence by $ECS(x,i)$.

Proof (of the theorem):
Towards a contradiction, assume there is a Turing machine M with space bound

$$s(n) \in o(\log\log(n)) \setminus O(1).$$

This means $s: \mathbb{N} \to \mathbb{N}$ is unbounded ($\forall m \in \mathbb{N}\ \exists n \in \mathbb{N}: s(n) > m$).

The number of small configurations on inputs of length n is

$$\leq |Q| \, |\Gamma|^{s(n)} \cdot (s(n)+2)$$

We have

$$|Q| \, |\Gamma|^{s(n)} \cdot (s(n)+2) \leq c^{s(n)} \quad \text{for large enough } n,$$

where c is a constant depending only on $|Q|$ and $|\Gamma|$.
This holds as $s(n)$ is unbounded.

- In an extended crossing sequence, no small configuration may appear twice in the same direction.
Otherwise, a (large) configuration of M would appear twice in the computation, and as M is deterministic it would be stuck in an infinite loop (M is assumed to halt)
Thus, there are at most

$$\underbrace{(c^{s(n)}+1)^{c^{s(n)}}}_{\text{left}} \cdot \underbrace{(c^{s(n)}+1)^{c^{s(n)}}}_{\text{right}}$$

$$= (c^{s(n)}+1)^{2 \cdot c^{s(n)}}$$

$$\leq 2^{2^{d s(n)}}$$

different extended crossing sequences on inputs of length n, where $d > 0$ is a constant.

- For large enough n_0,

$$s(n) < \frac{1}{2d} \log \log n \quad \text{for all } n \geq n_0.$$

Hence,

$$2^{2^{d\,s(n)}} < 2^{2^{d \cdot \frac{1}{2d} \log\log n}} = \left(2^{2^{\log\log n}}\right)^{\frac{1}{2}} = n^{\frac{1}{2}} \leq \frac{n}{2}$$

- Choose s_0 so that

$$s_0 > \max\{s(n) \mid 0 \leq n \leq n_0\}$$

and so that there is an input x with

$$\text{space}_M(x) = s_0.$$ < Such an s_0 exists because $s(n)$ is unbounded.

Let x be the shortest input with

$$s_0 = \text{space}_M(x).$$

- By the definition of s_0, we have

$$|x| > n_0,$$

for otherwise $\text{space}_M(x) = s_0 > s(|x|) \geq \text{space}_M(x)$. $\lightning$ (space bound)

Hence, by the definition of n_0 the number of crossing sequences is $< \frac{|x|}{2}$.

This means there are distinct positions $i < j < k$ with

$$ECS(x,i) = ECS(x,j) = ECS(x,k).$$

Indeed, if there were at most two positions for each extended crossing sequence, the length of x would be bounded by $|x| < 2 \cdot \frac{|x|}{2} = |x|$. $\lightning$

- We shorten the input by gluing the crossing sequences either at i and j or at j and k.

On at least one of the two new inputs,
M will use the same space.
Why? Every small configuration x
appears in at least one of the shortened strings.
If x was assumed to be the shortest string
with that space usage. □

Claim:

$L := \{ bin(0) \# bin(1) \# \dots \# bin(n) \mid n \in \mathbb{N} \}$

is
↳ not regular
↳ in $DSPACE(\log \log(n))$.

The above theorem can be phrased as:
"If M runs in $O(\log \log(n))$ space,
then M accepts a regular language".

This relationship is non-trivial and relies on
the following.

Theorem: $DSPACE(O(1)) = REG$.

The inclusion from right to left ($\supseteq$) is immediate.
For the reverse inclusion, the challenge is to mimic the behavior of a TM
by reading every letter only once
(the TM may visit the letter several times).